

Equivalent to Ref. 3

(19) 日本国特許庁(JP)

(12) 公表特許公報(A)

(11) 特許出願公表番号

特表2005-507100

(P2005-507100A)

(43) 公表日 平成17年3月10日(2005.3.10)

(51) Int. Cl.⁷

G06F 12/00

F I

G06F 12/00

5 3 3 F

G06F 12/00

5 1 7

テーマコード (参考)

5 B 0 8 2

審査請求 未請求 予備審査請求 有 (全 98 頁)

(21) 出願番号 特願2003-500739 (P2003-500739)
 (86) (22) 出願日 平成14年5月24日 (2002.5.24)
 (85) 翻訳文提出日 平成15年11月21日 (2003.11.21)
 (86) 国際出願番号 PCT/US2002/016481
 (87) 国際公開番号 W02002/097623 Reference 3
 (87) 国際公開日 平成14年12月5日 (2002.12.5)
 (31) 優先権主張番号 09/866, 143
 (32) 優先日 平成13年5月25日 (2001.5.25)
 (33) 優先権主張国 米国 (US)

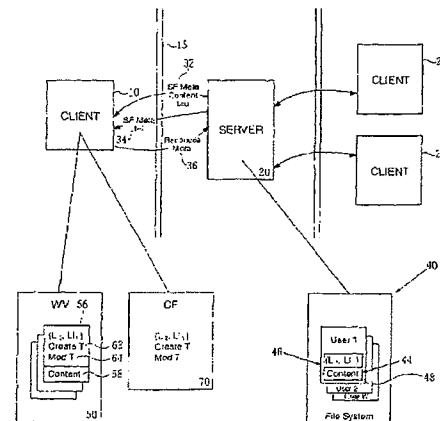
(71) 出願人 502303739
 オラクル・インターナショナル・コーポレーション
 アメリカ合衆国、94065 カリフォルニア州、レッドウッド・ショアーズ、オラクル・パークウェイ、500
 (74) 代理人 100064746
 弁理士 深見 久郎
 (74) 代理人 100085132
 弁理士 森田 俊雄
 (74) 代理人 100083703
 弁理士 仲村 義平
 (74) 代理人 100096781
 弁理士 堀井 豊

最終頁に続く

(54) 【発明の名称】 ネットワークファイルシステムのための管理および同期化アプリケーション

(57) 【要約】

ファイルを管理するためにアプリケーションが提供される。ファイルシステムについての情報が比較ファイルへマッピングされる。ファイルシステムから作業バージョンが作成される。アプリケーションは作業バージョンに対して1つ以上の動作が実行されたことを識別する。識別された動作は同期化動作中にファイルシステムに対して再現される。



【特許請求の範囲】**【請求項 1】**

ファイルを管理するための方法であって、
ファイルシステムについての情報を比較ファイルへマッピングするステップを含み、情報は、第 1 の位置に位置するファイルシステム内の少なくとも第 1 の項目を識別し、第 1 の位置は第 1 の位置情報により識別可能であり、前記方法はさらに、
ファイルシステムの一部の作業バージョンを作成するステップを含み、作業バージョンは第 1 の項目に対応する少なくとも第 1 の作業項目を含み、第 1 の作業項目は最初、第 1 の位置情報により識別可能な第 2 の位置に位置しており、前記方法はさらに、
比較ファイル内のおよび作業バージョンについての情報を用いて、第 1 の作業項目が第 2 の位置情報により識別可能な第 3 の位置に位置しているかどうかをその後判断するステップを含み、第 2 の位置情報は第 1 の位置情報とは異なる、方法。 10

【請求項 2】

第 1 の作業項目が第 3 の位置に位置する場合、ファイルシステム内の第 1 の項目を、第 2 の位置情報により識別可能な第 4 の位置へ移動させるステップを含む、請求項 1 に記載の方法。

【請求項 3】

比較ファイル内のおよび作業バージョンについての情報を用いて、第 1 の作業項目が第 2 の位置情報により識別可能な第 3 の位置に位置しているかどうかをその後判断するステップは、第 1 の作業項目が新しい名前を有しているかどうかを判断するステップを含む、請求項 1 に記載の方法。 20

【請求項 4】

ファイルシステムの一部の作業バージョンを作成するステップは、第 1 の作業項目の作成時刻を比較ファイルに記録するステップを含む、請求項 3 に記載の方法。

【請求項 5】

比較ファイル内のおよび作業バージョンについての情報を用いて、第 1 の作業項目が第 2 の位置情報により識別可能な第 3 の位置に位置しているかどうかをその後判断するステップは、作業バージョン内の第 1 の作業項目の位置を、作成時刻を用いて特定するステップを含む、請求項 3 に記載の方法。

【請求項 6】

第 1 の作業項目が作業バージョンから省略されている場合、第 1 の項目を削除するようファイルシステムに信号を送るステップをさらに含む、請求項 1 に記載の方法。 30

【請求項 7】

第 1 の作業項目の作成時刻を用いて、第 1 の作業項目が作業バージョンから削除されたと判断するステップをさらに含む、請求項 6 に記載の方法。

【請求項 8】

ファイルシステムの一部の作業バージョンを作成するステップは、第 1 の作業項目の修正時刻を比較ファイルに記録するステップを含む、請求項 4 に記載の方法。

【請求項 9】

作業バージョンの作成後に第 1 の作業項目が編集されたかどうかを判断するステップをさらに含む、請求項 1 に記載の方法。 40

【請求項 10】

作業バージョンの作成後に第 1 の作業項目が編集されたかどうかを、第 1 の作業項目の修正時刻を用いて判断するステップをさらに含む、請求項 8 に記載の方法。

【請求項 11】

第 1 の作業項目のその後の修正時刻が、第 1 の作業項目についての記録された修正時刻と異なるかどうかを判断するステップをさらに含む、請求項 10 に記載の方法。

【請求項 12】

ファイルを管理するための方法であって、
ファイルシステムについての情報を比較ファイルへマッピングするステップと、 50

ファイルシステムの一部の作業バージョンを作成するステップと、比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップとを含む、方法。

【請求項13】

第1の作業項目がかつて第2の作業項目からコピーされたと判断した後で、ファイルシステムの一部に、第1の作業項目を含めさせるステップをさらに含む、請求項12に記載の方法。

【請求項14】

第2の作業項目は、ファイルシステムの一部における第1の項目から生じている、請求項12に記載の方法。 10

【請求項15】

作業バージョンが作成された後で、第2の作業項目が新規として作成される、請求項13に記載の方法。

【請求項16】

第1の作業項目の内容は第2の作業項目の内容と異なっている、請求項13に記載の方法。

【請求項17】

作業バージョンを作成するステップは、第2の作業項目について作成時刻および修正時刻の双方を比較ファイルに記録するステップを含む、請求項14に記載の方法。 20

【請求項18】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップは、第1の作業項目についての作成時刻および修正時刻を識別するステップを含む、請求項17に記載の方法。

【請求項19】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップは、第1の作業項目の修正時刻が第1の作業項目の作成時刻の前であることを検出するステップを含む、請求項18に記載の方法。 30

【請求項20】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップは、第1の作業項目の修正時刻を第2の作業項目の修正時刻と整合させるステップを含む、請求項19に記載の方法。

【請求項21】

第1の作業項目が第2の作業項目からコピーされた後で編集されたかどうかを判断するステップをさらに含む、請求項12に記載の方法。

【請求項22】

第1の作業項目の作成時刻が比較ファイルにより識別された全項目の作成時刻と異なる場合、および第1の作業項目についての修正時刻が第1の作業項目についての作成時刻よりも大きいかまたはそれと等しい場合、第1の作業項目は新規であると判断するステップをさらに含む、請求項12に記載の方法。 40

【請求項23】

第1の作業項目が新規であると判断した後で、ファイルシステムの一部に、第1の作業項目を含めさせるステップをさらに含む、請求項22に記載の方法。

【請求項24】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップは、第1の作業項目の内容を第2の作業項目の内容と比較するス 50

トップを含む、請求項 1 2 に記載の方法。

【請求項 2 5】

ファイルを管理するための方法であって、
ファイルシステムについての情報を比較ファイルへマッピングするステップを含み、情報は、第 1 の位置に位置するファイルシステム内の少なくとも第 1 の項目を識別し、第 1 の位置は第 1 の位置情報により識別可能であり、前記方法はさらに、
ファイルシステムの一部の作業バージョンを作成するステップを含み、作業バージョンは第 1 の項目から生じる少なくとも第 1 の作業項目を含み、第 1 の作業項目は最初、第 2 の位置に位置しており、第 2 の位置は第 1 の位置情報により識別可能であり、前記方法はさらに、
比較ファイル内のおよび作業バージョンについての情報を用いて、2 つの動作のうちの少なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップを含み、動作は、第 1 の作業項目についての第 1 の位置情報の変更と、第 1 の作業項目の内容の編集とを含む、方法。

10

【請求項 2 6】

ファイルシステムの一部の作業バージョンを作成するステップは、第 1 の作業項目についての初期修正時刻を比較ファイルに記録するステップを含み、初期修正時刻は第 1 の作業項目が編集または作成された最後の時点を記録する、請求項 2 5 に記載の方法。

【請求項 2 7】

第 1 の作業項目の内容を編集する動作を検出するステップは、初期修正時刻が変更されたかどうかをその後判断するステップを含む、請求項 2 6 に記載の方法。

20

【請求項 2 8】

ファイルシステムの第 1 の項目に、第 1 の作業項目の編集された内容を含めさせるステップをさらに含む、請求項 2 7 に記載の方法。

【請求項 2 9】

ファイルシステムの作業バージョンを作成するステップは、第 1 の作業項目についての作成時刻を記録するステップを含む、請求項 2 5 に記載の方法。

【請求項 3 0】

比較ファイル内のおよび作業バージョンについての情報を用いて、2 つの動作のうちの少なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップは、作業バージョン内の第 1 の作業項目の位置を、作成時刻を用いて特定するステップを含む、請求項 2 9 に記載の方法。

30

【請求項 3 1】

ファイルを管理するための方法であって、
ファイルシステムについての情報を比較ファイルへマッピングするステップを含み、情報は、第 1 の位置に位置するファイルシステム内の少なくとも第 1 の項目を識別し、第 1 の位置は第 1 の位置情報により識別可能であり、前記方法はさらに、
ファイルシステムの一部の作業バージョンを作成するステップを含み、作業バージョンは第 1 の項目から生じる少なくとも第 1 の作業項目を含み、第 1 の作業項目は最初、第 2 の位置に位置しており、第 2 の位置は第 1 の位置情報により識別可能であり、前記方法はさらに、
比較ファイル内のおよび作業バージョンについての情報を用いて、複合動作が第 1 の作業項目に対して実行されたかどうかを判断するステップを含み、複合動作は、第 1 の作業項目についての第 1 の位置情報の変更、第 1 の作業項目からの第 1 の作業コピーの作成、および第 1 の作業項目の内容の編集からなる一組の動作からの少なくとも 2 つの連続する動作を含む、方法。

40

【請求項 3 2】

第 1 の項目が第 1 の作業項目に整合するように、ファイルシステムの一部に、作業バージョンに対して実行された動作によりなされた変更を取入れさせるステップをさらに含む、請求項 3 1 に記載の方法。

50

【請求項 3 3】

少なくとも 2 つの複合動作のうちの 1 つが第 1 の作業項目から第 1 の作業コピーを作成するために、作業バージョン内の選択された作業項目がかつて第 1 の作業項目からコピーされたかどうかを判断するステップをさらに含む、請求項 3 1 に記載の方法。

【請求項 3 4】

コンピュータシステムであって、

第 2 のコンピュータシステム上に位置するファイルシステムの一部の作業バージョンを記憶するメモリと、

作業バージョン内の第 1 の作業項目の少なくとも作成時刻をマークするプロセッサとを含み、第 1 の作業項目はファイルシステムの一部の第 1 の項目から生じ、プロセッサはその後作成時刻を用いて第 1 の作業項目に対して実行された動作を検出する、コンピュータシステム。

10

【請求項 3 5】

ネットワークを通して第 2 のコンピュータと通信をやり取りするためのネットワークインターフェイスをさらに含み、通信は作業バージョンを作成するためのものである、請求項 3 4 に記載のコンピュータシステム。

【請求項 3 6】

プロセッサは作成時刻を用いて、第 1 の作業項目の位置がその後変更されたかどうかを判断する、請求項 3 4 に記載のコンピュータシステム。

【請求項 3 7】

プロセッサは作成時刻を用いて、多数の動作が第 1 の作業項目に対して実行されたかどうかを判断する、請求項 3 4 に記載のコンピュータシステム。

20

【請求項 3 8】

プロセッサは最初、作成時刻とともに修正時刻をマークし、修正時刻は第 1 の項目がその後編集されるときに変更される、請求項 3 4 に記載のコンピュータシステム。

【請求項 3 9】

プロセッサは作成時刻を修正時刻と比較して、第 1 の作業項目に対して実行された動作を識別する、請求項 3 8 に記載のコンピュータシステム。

【請求項 4 0】

修正時刻がその後、最初にマークされたものから変更されている場合、プロセッサは第 1 の作業項目の内容が編集されたことを検出する、請求項 3 8 に記載のコンピュータシステム。

30

【請求項 4 1】

プロセッサは、第 2 の作業項目がかつて第 1 の作業項目からコピーされたと判断するために、第 2 の作業項目が第 1 の作業項目の修正時刻を有するかどうかを検出する、請求項 3 8 に記載のコンピュータシステム。

【請求項 4 2】

プロセッサは、第 1 の作業項目の作成時刻を検出して、第 1 の作業項目が新しい位置へ移動されたと判断し、プロセッサは、第 1 の作業項目の修正時刻が変更されたことを検出して、第 1 の作業項目が編集もされたと判断する、請求項 3 8 に記載のコンピュータシステム。

40

【請求項 4 3】

コンピュータシステムであって、

第 2 のコンピュータ上でアクセス可能なファイルシステムの一部の作業バージョンを記憶するメモリを含み、作業バージョンは複数の作業項目を含み、メモリは複数の作業項目についての情報を含む比較ファイルも記憶しており、前記コンピュータシステムはさらに、作業バージョン内の第 1 の組の作業項目についての少なくとも第 1 の時刻値を、第 1 の組の作業項目が最初にメモリに記憶されたときに記録するよう構成されたプロセッサを含み、プロセッサはその後作業バージョン内の第 2 の作業項目の第 1 の時刻値を識別し、プロセッサは第 2 の作業項目の第 1 の時刻値を第 1 の組の作業項目の第 1 の時刻値と比較して

50

、第2の作業項目に影響を与える作業バージョンに対して実行された動作を検出する、コンピュータシステム。

【請求項44】

ネットワークを通して第2のコンピュータと通信をやり取りするためのネットワークインターフェイスをさらに含み、通信はファイルシステムから作業バージョンを作成するためのものである、請求項43に記載のコンピュータシステム。

【請求項45】

プロセッサは、第1の組の作業項目の第1の時刻値を比較ファイルに記録する、請求項43に記載のコンピュータシステム。

【請求項46】

比較ファイルは、ファイルシステムの一部における複数の項目についての情報も含み、複数の作業項目の少なくともいくつかは、ファイルシステムの一部における複数の項目から生じている、請求項44に記載のコンピュータシステム。

【請求項47】

第1の作業項目の第1の時刻値は、それらの項目についての作成時刻に対応している、請求項43に記載のコンピュータシステム。

【請求項48】

検出された動作は、第1の組の作業項目の第1の時刻値が記録された後で第2の項目を作成するステップを含む、請求項43に記載のコンピュータシステム。

【請求項49】

検出された動作は、第1の組の作業項目のうちの1つのコピーとして第2の項目を作成するステップを含む、請求項48に記載のコンピュータシステム。

【請求項50】

第1の組の作業項目が最初にメモリに記憶されたときに、プロセッサは第1の組の作業項目の各々についての少なくとも第2の時刻値を記録し、第2の時刻値は、対応する項目の内容が最後に変更または作成された修正時刻に対応する、請求項43に記載のコンピュータシステム。

【請求項51】

プロセッサは、第2の作業項目の第2の時刻値を識別し、第2の作業項目の第1の時刻値および第2の時刻値を、第1の組の作業項目の第1の時刻値および第2の時刻値と比較して、動作を識別する、請求項50に記載のコンピュータシステム。

【請求項52】

第1の組の作業項目はファイルシステムの一部における対応する項目から生じており、プロセッサは動作を、作業バージョンが受取られた後での第2の作業項目の作成として検出する、請求項51に記載のコンピュータシステム。

【請求項53】

プロセッサは動作を、第1の組の作業項目のうちの1つからの第2の作業項目の作成として検出する、請求項52に記載のコンピュータシステム。

【請求項54】

プロセッサは動作を、新規項目としての第2の作業項目の作成として検出する、請求項53に記載のコンピュータシステム。

【請求項55】

第1の作業項目の第1の時刻値は、それらの項目についての作成時刻に対応しており、第2の作業項目についてのおよび第1の組の作業項目についての第2の時刻値は、作業バージョン内のそれぞれの作業項目についての修正時刻に対応している、請求項53に記載のコンピュータシステム。

【請求項56】

第2の時刻値が第1の組の作業項目のうちの1つの第1の時刻値と整合する場合、動作は、第1の組の作業項目のうちの1つからの第2の作業項目の作成として検出される、請求項55に記載のコンピュータシステム。

10

20

30

40

50

【請求項 5 7】

第 3 の作業項目を作成する第 1 の動作と、第 3 の作業項目のコピーとして第 2 の作業項目を作成する第 2 の動作とを含む多数の動作が検出される、請求項 5 5 に記載のコンピュータシステム。

【請求項 5 8】

第 3 の作業項目の第 1 の時刻値が第 1 の組の作業項目のうちの 1 つの第 1 の時刻値と整合しない場合、プロセッサは第 1 の動作を、新規項目としての第 3 の作業項目の作成として検出する、請求項 5 7 に記載のコンピュータシステム。

【請求項 5 9】

第 2 の作業項目の第 2 の時刻値が第 2 の作業項目の第 1 の時刻値の前である場合、および第 2 の作業項目の第 2 の時刻値が第 3 の作業項目の第 1 の時刻値に整合する場合、プロセッサは第 2 の動作を、第 3 の作業項目のコピーとしての第 2 の作業項目の作成として検出する、請求項 5 8 に記載のコンピュータシステム。 10

【請求項 6 0】

コンピュータシステムであって、
ファイルシステムを記憶するサーバと、
ファイルシステムに結合可能で、ファイルシステムの少なくとも一部を検索するクライアントとを含み、ファイルシステムの一部は、作業バージョンとしてクライアント上で利用可能となっており、作業バージョンは最初、複数の作業項目を含んでおり、作業項目の各々はファイルシステム内の対応する項目から生じており、前記コンピュータシステムはさらに、
作業バージョンについての初期情報を記録し、その後、作業バージョンに対して実行される 2 つ以上の動作を検出する処理資源を含み、動作は、作業バージョンへの項目の追加、作業バージョン内の項目の編集、作業バージョンからの項目の削除、および作業バージョン内の項目の新しい位置への移動からなる動作の群から選択される、コンピュータシステム。 20

【請求項 6 1】

処理資源は、作業バージョンに対して実行された動作を反映するようファイルシステムの少なくとも一部を変更するようにファイルシステムへ信号を送る、請求項 6 0 に記載のコンピュータシステム。 30

【請求項 6 2】

作業バージョンへの項目の追加という動作は、作業バージョン内の第 1 の作業項目からの第 2 の作業項目のコピーを含む、請求項 6 0 に記載のコンピュータシステム。

【請求項 6 3】

作業バージョンへの項目の追加という動作は、新規項目としての第 2 の作業項目の作成を含む、請求項 6 0 に記載のコンピュータシステム。

【請求項 6 4】

作業ファイルについて記録された初期情報は、作業バージョン内の複数の作業項目についての第 1 の時刻値を含み、第 1 の時刻値は、複数の作業項目内の別の作業項目からの、またはサーバ上の別の項目からの各作業項目の起点を示す、請求項 6 0 に記載のコンピュータシステム。 40

【請求項 6 5】

第 1 の時刻値は、その項目がクライアント上で作成されたときに対応する作成時刻である、請求項 6 4 に記載のコンピュータシステム。

【請求項 6 6】

第 1 の時刻値は、その項目が変更または作成されたときの最後の時点を記録する修正時刻である、請求項 6 4 に記載のコンピュータシステム。

【請求項 6 7】

処理資源はクライアント上に位置している、請求項 6 0 に記載のコンピュータシステム。

【請求項 6 8】

処理資源はサーバ上に位置している、請求項 60 に記載のコンピュータシステム。

【請求項 69】

処理資源はクライアントとサーバとの間で共有されている、請求項 59 に記載のコンピュータシステム。

【請求項 70】

ファイルシステムは複数のクライアントで共有されており、複数のクライアントの各々はファイルシステムを変更可能であり、クライアントが作業バージョンをファイルシステムと比較するよう要求を行なうと、ファイルシステムの変更されたバージョンがクライアントへ信号で送られる、請求項 60 に記載のコンピュータシステム。

【請求項 71】

サーバはファイルシステムの変更されたバージョンをクライアントへ信号で送り、処理資源はクライアント上にあって、処理資源が作業バージョンについての初期情報を記録した後でファイルシステムに対して実行された 1 つ以上の動作を検出する、請求項 70 に記載のコンピュータシステム。

10

【請求項 72】

処理資源は、ファイルシステムの変更されたバージョンを作業バージョンについて記録された初期情報と比較して、ファイルシステムに対して実行された 1 つ以上の動作を検出する、請求項 71 に記載のコンピュータシステム。

【請求項 73】

処理資源は、1 つ以上の動作の結果変更または作成された各作業バージョン項目を用いてファイルシステムを更新するかどうかを選択するよう、ユーザへ信号を送る、請求項 72 に記載のコンピュータシステム。

20

【請求項 74】

処理資源は、ファイルシステムに対して実行された 1 つ以上の動作の結果変更または作成された、変更されたファイルシステム内の各項目を用いてファイルシステムを更新するかどうかを選択するよう、ユーザへ信号を送る、請求項 73 に記載のコンピュータシステム。

【請求項 75】

処理資源は、作業バージョンの第 1 の項目と変更されたファイルシステムの第 1 の項目との間で選択するようユーザへ信号を送り、作業バージョンの第 1 の項目は動作の群からの第 1 の動作の対象となっており、ファイルシステムの第 1 の項目は動作の群からの第 2 の動作の対象となっており、作業バージョンの第 1 の項目および変更されたファイルシステムの第 1 の項目は各々、ファイルシステム内の第 1 の項目から生じており、第 1 の動作は第 2 の動作と異なっている、請求項 74 に記載のコンピュータシステム。

30

【請求項 76】

ファイルを管理するための命令を搬送するコンピュータ可読媒体であって、命令は、ファイルシステムについての情報を比較ファイルへマッピングするステップを行なうための命令を含み、情報は、第 1 の位置に位置するファイルシステム内の少なくとも第 1 の項目を識別し、第 1 の位置は第 1 の位置情報により識別され、前記命令はさらに、ファイルシステムの一部の作業バージョンを作成するステップを行なうための命令を含み、作業バージョンは第 1 の項目から生じる少なくとも第 1 の作業項目を含み、第 1 の作業項目は最初、第 1 の位置情報によって第 2 の位置に位置しており、前記命令はさらに、比較ファイル内のおよび作業バージョンについての情報を用いて、第 1 の作業項目が第 2 の位置情報によって識別される第 3 の位置に位置しているかどうかを判断するステップを行なうための命令を含み、第 2 の位置情報は第 1 の位置情報とは異なる、コンピュータ可読媒体。

40

【請求項 77】

第 1 の作業項目が第 3 の位置に位置する場合、コンピュータ可読媒体の命令は、ファイルシステム内の第 1 の項目を、第 2 の位置情報によって識別される第 4 の位置へ移動させるための命令を含む、請求項 76 に記載のコンピュータ可読媒体。

50

【請求項 7 8】

比較ファイル内のおよび作業バージョンについての情報を用いて、第 1 の作業項目がその後、第 2 の位置情報により識別可能な第 3 の位置に位置しているかどうかを判断するステップは、第 1 の作業項目が新しい名前を有しているかどうかを判断するための命令を含む、請求項 7 6 に記載のコンピュータ可読媒体。

【請求項 7 9】

ファイルシステムの一部の作業バージョンを作成するステップは、第 1 の作業項目の作成時刻を比較ファイルに記録するための命令を含む、請求項 7 8 に記載のコンピュータ可読媒体。

【請求項 8 0】

比較ファイル内のおよび作業バージョンについての情報を用いて、第 1 の作業項目がその後、第 2 の位置情報により識別可能な第 3 の位置に位置しているかどうかを判断するステップは、作業バージョン内の第 1 の作業項目の位置を、作成時刻を用いて特定するための命令を含む、請求項 7 8 に記載のコンピュータ可読媒体。

10

【請求項 8 1】

第 1 の作業項目が作業バージョンから省略されている場合、第 1 の項目を削除するようファイルシステムに信号を送るための命令をさらに含む、請求項 7 6 に記載のコンピュータ可読媒体。

【請求項 8 2】

第 1 の作業項目の作成時刻を用いて、第 1 の作業項目が作業バージョンから削除されたと判断するための命令をさらに含む、請求項 8 1 に記載のコンピュータ可読媒体。

20

【請求項 8 3】

ファイルシステムの作業バージョンを作成するステップは、第 1 の作業項目の修正時刻を比較ファイルに記録するための命令を含む、請求項 7 9 に記載のコンピュータ可読媒体。

【請求項 8 4】

作業バージョンの作成後に第 1 の作業項目が編集されたかどうかを判断するための命令をさらに含む、請求項 7 6 に記載のコンピュータ可読媒体。

【請求項 8 5】

作業バージョンの作成後に第 1 の作業項目が編集されたかどうかを、第 1 の作業項目の修正時刻を用いて判断するための命令をさらに含む、請求項 8 3 に記載のコンピュータ可読媒体。

30

【請求項 8 6】

第 1 の作業項目の修正時刻を用いるステップは、第 1 の作業項目のその後の修正時刻が第 1 の作業項目についての記録された修正時刻と異なるかどうかを判断するための命令を含む、請求項 8 5 に記載のコンピュータ可読媒体。

【請求項 8 7】

異なるコンピュータ上のファイルを管理するための命令を搬送するコンピュータ可読媒体であって、命令は、

ファイルシステムについての情報を比較ファイルへマッピングするステップと、

ファイルシステムの一部の作業バージョンを作成するステップと、

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップとを行なうための命令を含む、コンピュータ可読媒体。

40

【請求項 8 8】

第 1 の作業項目がかつて第 2 の作業項目からコピーされたと判断した後でファイルシステムに第 1 の作業項目を含めさせるための命令をさらに含む、請求項 8 7 に記載のコンピュータ可読媒体。

【請求項 8 9】

第 2 の作業項目は、ファイルシステムにおける第 1 の項目から生じている、請求項 8 7 に記載のコンピュータ可読媒体。

50

【請求項 9 0】

第 1 の作業項目の内容は第 2 の作業項目の内容と異なっている、請求項 8 8 に記載のコンピュータ可読媒体。

【請求項 9 1】

作業バージョンを作成するステップは、第 2 の作業項目について作成時刻および修正時刻の双方を比較ファイルに記録するための命令を含む、請求項 8 9 に記載のコンピュータ可読媒体。

【請求項 9 2】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップは、第 1 の作業項目についての作成時刻および修正時刻を識別するための命令を含む、請求項 9 1 に記載のコンピュータ可読媒体。

10

【請求項 9 3】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップは、第 1 の作業項目の修正時刻が第 1 の作業項目の作成時刻の前であることを検出するための命令を含む、請求項 9 2 に記載のコンピュータ可読媒体。

【請求項 9 4】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップは、第 1 の作業項目の修正時刻を第 2 の作業項目の修正時刻と整合させるための命令を含む、請求項 9 3 に記載のコンピュータ可読媒体。

20

【請求項 9 5】

第 1 の作業項目が第 2 の作業項目からコピーされた後で編集されたかどうかを判断するステップを行なうための命令をさらに含む、請求項 8 9 に記載のコンピュータ可読媒体。

【請求項 9 6】

第 1 の作業項目の作成時刻が比較ファイルにより識別された全項目の作成時刻と異なる場合、および第 1 の作業項目についての修正時刻が第 1 の作業項目についての作成時刻よりも大きいまたはそれと等しい場合、第 1 の作業項目は新規ファイルであると判断するステップを行なうための命令をさらに含む、請求項 8 7 に記載のコンピュータ可読媒体。

30

【請求項 9 7】

第 1 の作業項目が新規であると判断した後でファイルシステムに第 1 の作業項目を含めさせるステップを行なうための命令をさらに含む、請求項 9 6 に記載のコンピュータ可読媒体。

【請求項 9 8】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップは、第 1 の作業項目の内容を第 2 の作業項目の内容と比較するための命令を含む、請求項 8 2 に記載のコンピュータ可読媒体。

【請求項 9 9】

ファイルを管理するためのコンピュータ可読媒体であって、コンピュータ可読媒体は、ファイルシステムについての情報を比較ファイルへマッピングするステップを行なうための命令を含む命令を搬送しており、その情報は、第 1 の位置に位置するファイルシステム内の少なくとも第 1 の項目を識別し、第 1 の位置は第 1 の位置情報によって識別され、前記コンピュータ可読媒体はさらに、

40

ファイルシステムの一部の作業バージョンを作成するステップを行なうための命令を含む命令を搬送しており、作業バージョンは第 1 の項目から生じる少なくとも第 1 の作業項目を含み、第 1 の作業項目は最初、第 2 の位置に位置しており、第 2 の位置は第 1 の位置情報によって識別され、前記コンピュータ可読媒体はさらに、

比較ファイル内のおよび作業バージョンについての情報を用いて、2 つの動作のうちの少

50

なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップを行なうための命令を含む命令を搬送しており、動作は、第1の作業項目についての第1の位置情報の変更と、第1の作業項目の内容の編集とを含む、コンピュータ可読媒体。

【請求項100】

ファイルシステムの一部の作業バージョンを作成するステップは、第1の作業項目についての初期修正時刻を比較ファイルに記録するための命令を含み、初期修正時刻は第1の作業項目が編集または作成された最後の時点を記録する、請求項99に記載のコンピュータ可読媒体。

【請求項101】

比較ファイル内のおよび作業バージョンについての情報を用いて、2つの動作のうちの少なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップは、初期修正時刻が変更されたかどうかをその後判断するための命令を含む、請求項100に記載のコンピュータ可読媒体。

10

【請求項102】

ファイルシステムに第1の作業項目の編集された内容を含めさせるステップを行なうための命令をさらに含む、請求項101に記載のコンピュータ可読媒体。

【請求項103】

ファイルシステムの一部の作業バージョンを作成するステップは、第1の作業項目についての作成時刻を記録するための命令を含む、請求項99に記載のコンピュータ可読媒体。

【請求項104】

比較ファイル内のおよび作業バージョンについての情報を用いて、2つの動作のうちの少なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップは、作業バージョン内の第1の作業項目の位置を、作成時刻を用いて特定するための命令を含む、請求項103に記載のコンピュータ可読媒体。

20

【請求項105】

ファイルを管理するための命令を搬送するコンピュータ可読媒体であって、命令は、ファイルシステムについての情報を比較ファイルへマッピングするステップを行なうための命令を含み、情報は、第1の位置に位置するファイルシステム内の少なくとも第1の項目を識別し、第1の位置は第1の位置情報によって識別され、前記命令はさらに、ファイルシステムの一部の作業バージョンを作成するステップを行なうための命令を含み、作業バージョンは第1の項目から生じる少なくとも第1の作業項目を含み、第1の作業項目は最初、第2の位置に位置しており、第2の位置は第1の位置情報によって識別され、前記命令はさらに、

30

比較ファイル内のおよび作業バージョンについての情報を用いて、複合動作が第1の作業項目に対して実行されたかどうかを判断するステップを行なうための命令を含み、複合動作は、第1の作業項目についての第1の位置情報の変更、第1の作業項目からの第1の作業コピーの作成、および第1の作業項目の内容の編集からなる一組の動作からの少なくとも2つの連続する動作を含む、コンピュータ可読媒体。

【請求項106】

複合動作が実行されたと判断された場合、ファイルシステム内の第1の項目を変更するためにどちらの動作が第1の作業項目に対して実行されたかを示すための命令をさらに含む、請求項105に記載のコンピュータ可読媒体。

40

【請求項107】

少なくとも2つの複合動作のうちの1つが第1の作業項目から第1の作業コピーを作成するために、作業バージョン内の選択された作業項目がかつて第1の作業項目からコピーされたかどうかを判断するための命令をさらに含む、請求項105に記載のコンピュータ可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

50

発明の分野

この発明はネットワークファイルを管理するためのアプリケーションに関する。特に、この発明の実施例は、同期化のためにファイルシステムへの変更を検出することに関係する。

【背景技術】

【0002】

発明の背景

多数のファイルシステムが全く同じ内容を含んでいる場合、それらのファイルシステムは「同期している」と言われる。ファイルシステムを同期した状態に保つために、同期化アプリケーションがファイルシステム間の違いを検出し、次にその違いをなくすよう動作を実行する。通常、同期化アプリケーションは、同じファイルの異なるコピーにアクセスする必要がある、異なるコンピュータのファイルシステムを同期させるために使用される。互いに同期されるべき1組のファイルシステムはここで「同期化セット」と呼ばれる。所与の同期化セットに属する各ファイルシステムは「同期システム」と呼ばれる。

【0003】

同期化セットを同期させる行為は同期化動作と呼ばれる。各同期化動作中、同期化アプリケーションは通常、以前の同期化動作以降に任意の同期システムにおいていつ項目が削除されたかまたは追加されたかを検出しようとする。

【0004】

一般に、同期化アプリケーションは、以前の同期化動作以降に任意の同期システムに追加されていると検出された項目を、すべての同期システムへ追加しようとする。同様に、同期化アプリケーションは、以前の同期化動作以降に任意の同期システムから削除されていると検出された項目を、すべての同期システムから削除しようとする。

【発明の開示】

【発明が解決しようとする課題】

【0005】

通常、同期化アプリケーションは(1)他の項目のコピーである追加項目と(2)オリジナルとして作成された追加項目とを区別しない。加えて、改名された項目が変更された場合、元の項目が削除されて新規項目が追加されたように見える。したがって、元の項目がすべての同期システムにおいて削除されるかもしれず、新規の変更された項目がすべての同期システムに追加される。また、これに代えて、同期化アプリケーションは、変更された項目と変更されていない項目の双方をともに同じファイル内に置くかもしれない。

【0006】

現行の同期化手法の下では、ユーザがある項目に対して多数の動作を実行し、次にその項目を別のものと同期させようとする場合、アプリケーションはその項目を、変更されたかまたは新規であるとして検出する。アプリケーションは、その項目に対して実行された特定の動作を検出できないであろう。その結果、同期化動作が、あるファイルを別のものの代わりに削除したり、または、元のファイルが同じファイルシステム内にある状態で、変更されたファイルを記憶されるよう追加するかもしれない。

【0007】

同期化動作はしばしば、著しい量の資源消費を伴う。たとえば、同期化アプリケーションがある同期システムへの新規ファイルの追加を検出した場合、他のすべての同期システムへの新規ファイルの内容のデータ転送が必要とされる。ファイルが大きい場合、および/または同期システムの数が多い場合、資源消費はかなりのものとなり得る。

【0008】

現行の同期化手法での別の問題は、新規ファイルまたは置換されたファイルが、ファイルシステムに対するそれらの転送または再作成の前からのメタデータ情報を保持していないということである。このため、時刻T1で作成されたファイルが変更される場合、そのファイルが元々時刻T1で作成されたという事実は、同期化アプリケーションが、変更されたファイルを新規ファイルとして、元のファイルを削除されたファイルとして扱う場合に

10

20

30

40

50

よって損失されるであろう。

【課題を解決するための手段】

【0009】

発明の概要

この発明の実施例は、同期されるべき第1のファイルシステムに対して実行された1つ以上の動作を検出可能なアプリケーションを提供する。同期化アプリケーションは、第1のファイルシステムの検出された動作を用いて、第2のファイルシステムを更新する。

【0010】

この発明の一実施例は、項目のコピー、項目の移動、新規項目の作成、項目の削除、および項目の編集を含む、第1のファイルシステムにおける動作を検出可能である。この発明の一実施例は、第1のファイルシステムに対して実行された多数の動作も検出する。検出された動作は、同期化動作中、第2のファイルシステムに対して再現されてもよい。

10

【0011】

この発明の実施例で説明される同期化手法は、使用される他の同期化プロセスよりも、被るオーバーヘッドが少ない。さらに、この発明の実施例は、同期化されたファイルについてのメタデータ情報を保存する同期化手法を提供しており、これはそのような情報の損失を被る他の同期化プロセスとは対照的である。

【発明を実施するための最良の形態】

【0012】

この発明を添付図面の図において、限定のためではなく例示のために説明する。図中、同じ参照符号は同様の要素を指す。

20

【0013】

発明の詳細な説明

ファイルを管理するための方法および装置を記載する。以下の記載では、説明のため、この発明の完全な理解を提供するために多数の特定の詳細が述べられている。しかしながら、この発明がこれらの特定の詳細なしで実践されてもよいことは明らかである。他の場合、この発明を不必要に不明瞭にするのを避けるために、周知の構造および装置をブロック図の形で示す。

【0014】

提供される利点の中でも、この発明の実施例は、同期システムに属するファイル項目へ実行された特定の行為を識別する同期化動作が実行されることを可能にする。同期化動作中、特定の動作は他の同期システムの対応するファイル項目に対して再現される。その結果、ファイル項目が単にどこか変更または移動されただけである多くの場合において、そのようなファイル項目の完全な転送が回避され得る。結果として通信資源が節約される。加えて、他の同期システムの対応するファイル項目は、それらの項目についてのメタデータ情報を失うことなく更新され得る。

30

【0015】

A. 機能概要

一実施例では、1つ以上のクライアント端末がサーバ上のファイルシステムにアクセス可能である。1つ以上のファイルがファイルシステムからダウンロードされ、クライアント端末上で処理され得る。特に、ユーザはダウンロードされたファイルの内容に対して、文書の編集、項目の削除、ファイル用の新規文書の作成、ファイル内の項目の移動または項目のコピー、もしくはこれらの行為の組合せを含むある行為を実行してもよい。一実施例の下では、ダウンロードされたファイルに対して実行された行為を管理システムが検出する。管理システムは次に、ダウンロードされたファイルをファイルシステムの対応する部分と同期させる。

40

【0016】

ここに記載するように、ファイルシステムは、多数のクライアント用に多くのファイルを保持する管理システム全体の一部である。クライアントはファイルシステムの一部のみをダウンロードしてもよい。ファイルシステムの一部は、ディレクトリ、サブファイル、ア

50

アプリケーション、実行ファイル、文書、および異なるデータ種類の個々の資源などの項目を含んでいてもよい。

【0017】

ファイルシステムの一部がクライアントによってダウンロードされる場合、結果として生じるローカルファイルは作業バージョンと呼ばれる。作業バージョンは、クライアント用にダウンロードされるよう選択されたファイルシステムの一部から項目をコピーする。情報が、作業バージョンについての比較ファイル内に、それが作成される際に記録される。比較ファイルは、ダウンロードされたファイルシステムの一部についての情報も含んでいてもよい。この情報は、ファイル項目を識別するためにその後使用され得るメタデータ情報と、作業バージョンが作成された後でそれになされた修正とを含む。作業バージョンが修正された後、作業バージョンは、作業バージョンをダウンロードするために使用されたファイルシステムの一部と同期され得る。比較ファイル内に記録された情報を用いて、作業バージョンへなされた変更が検出される。

10

【0018】

ここで用いられるように、用語「項目」への言及は、ファイルシステム内で維持および／または管理可能なデータ構造を意味する。上述のように、項目は、ディレクトリ、ファイル、アプリケーション、実行ファイル、文書、および異なるデータ種類の個々の資源を含む。項目は特定のデータ種類の文書または資源を含んでいてもよい。たとえば、第1の項目はワープロ文書であってもよく、第2の項目はその文書を他の資源とともに記憶するフォルダであってもよい。

20

【0019】

一実施例の下では、比較ファイルに含まれる情報は主としてメタデータ情報である。メタデータ情報は、特定の項目についての位置情報、作成時刻、修正時刻、項目の大きさ、およびファイル名を含んでいてもよい。

【0020】

位置は、メモリアドレスおよびコンピュータ位置によって識別可能である。位置情報は、コンピュータ上の項目の記憶位置を識別するために使用可能なデータを指す。位置情報はファイル名または資源名を含んでいてもよい。位置情報は、メモリ内の特定の項目の位置を特定するファイル経路も含んでいてもよい。

【0021】

この発明の実施例は、ファイルを管理するシステムおよび方法を含む。一実施例によれば、第1の同期化動作の後、情報がファイルシステムから比較ファイルへマッピングされる。その情報は、第1の同期化動作後の第1の項目のステータスについての情報を含む。たとえば、比較ファイル内の情報は、第1の同期化動作の後、第1の項目が第1の位置に位置したことを示してもよい。ファイルシステム情報を比較ファイルへマッピングすることに加え、ファイルシステムの作業バージョンが作成される。初めに、作業バージョンは第1の項目が第1の位置にあることを示す。作業バージョンに含まれる情報は、第1の同期化動作後に第1のファイルのステータスへなされたあらゆる変更を反映するよう修正される。その後の第2の同期化動作中、ファイルシステムの作業バージョンに示された位置は、比較ファイルに示された位置と比較され、第1の同期化動作と第2の同期化動作の合間に第1の項目が移動されたかどうか判断される。

30

40

【0022】

一実施例では、ファイルシステム内の第1の項目は、新しい位置情報によって識別可能な新しい位置へ移動されてもよい。したがって、第1の作業項目が移動される場合、ファイルシステムは第1の作業項目を再作成しなくてもよい。むしろ、ファイルシステムは、第1の作業ファイルに対応する第1の項目を、ファイルシステム内の対応する位置へ移動させることができる。以前の同期化アプリケーションとは対照的に、この発明のそのような実施例は、第1の作業項目が、作業バージョン上で移動されたというだけの理由で、ファイルシステムの新しい部分として再作成されることを必要としない。その結果、第1の作業項目の内容のデータ転送が不必要であるため、通信資源が保存される。さらに、ファイ

50

ルシステム内の第1の項目についてのメタデータ情報が保存される。

【0023】

この発明の別の実施例は、ファイルシステムについての情報を比較ファイルへマッピングする。作業バージョンはファイルシステムの一部から作成される。同期化動作中、比較ファイルと作業バージョンとを用いて、最後の同期化動作以降に項目がコピーされたかどうか判断される。

【0024】

この発明の別の実施例は、ネットワーク管理アプリケーションを動作可能なコンピュータシステムを提供する。このコンピュータシステムは、第2のコンピュータと通信をやり取りするためのネットワークインターフェイスを含む。この通信は、第2のコンピュータ上でアクセス可能なファイルシステム部分の作業バージョンを作成することになっている。第1のコンピュータは、作業バージョンを記憶するメモリを含む。第1のコンピュータ上のプロセッサは、作業バージョン内の少なくとも1つの作業項目についての作成時刻を記録し、第1の作業項目はファイルシステムの第1の項目から生じている。プロセッサはその後、作成時刻を用いて、第1の作業項目に対して動作が実行されたかどうかを判断する。

【0025】

他の利点の中でも、この発明の実施例は、作業バージョンが作成された後である項目が新しい位置へ移動されたかどうか、作業バージョンが別のファイルから作成された際にある項目が作業バージョン内に存在する別の項目からコピーされたかどうか、または、ある項目が作業バージョンの作成後に作業バージョンに追加された項目からコピーされたかどうかを検出可能である。この発明の一実施例の下で検出可能な他の動作は、項目が編集されたかどうか、または作業バージョンから削除されたかどうかを含む。この発明の一実施例は、作業バージョン内の項目に対してまたは項目のために実行された多数の動作も検出可能である。

【0026】

結果として、作業バージョンを作成するために使用されたファイルシステムの一部は、作業バージョンにおけるその後の変更を反映するよう更新され得る。しかしながら、作業バージョン項目によってその後更新されるファイルシステム内の項目は、作業バージョン項目によってすべて置換されたりまたは再作成される必要はない。むしろ、同期化方法またはアプリケーションは、作業バージョンの対応する項目への変更、または作業バージョンへの項目の追加を反映するようにファイルシステムを更新可能である。別の利点は、作業バージョン内で動作されるファイルシステム項目がそれらの起源を追跡する情報を維持できるといことである。その結果、ファイルシステムは、作業バージョンに対して実行された動作のみを反映するよう更新可能である。

【0027】

B. システム概要

図1は、この発明の一実施例の下で、コンピュータ間で共有されるファイルを管理するためのシステムを示す。このシステムは、ネットワーク15を介してサーバ20へ結合された第1の端末10を含む。複数の他の端末25もネットワーク15を通してサーバ20へ結合されていてもよい。第1の端末10は、サーバ20と通信するクライアントとして動作されてもよい。一実施例では、サーバ20と共有されるファイル項目および資源を管理するために、クライアントアプリケーションが第1の端末10上で動作可能である。

【0028】

ユーザは第1の端末10を動作させて、1つ以上の資源および他の項目を含むファイルシステム40へサーバ20からアクセスしてもよい。ユーザがファイルシステム40の一部の借りられたまたは作業用のバージョン上で作業している間、ファイルシステム40の元々のバージョンはサーバ20上に残っていてもよい。図1によって説明されるようなシステムにより、ユーザは、遠隔サーバ20からアクセスされた項目に対して局所的に作業し、次に、第1の端末10上でそれらの項目へなされた変更を反映するよう、遠隔サーバ上

のファイルシステム 40 を更新することができる。

【0029】

一実施例では、端末 10 上のユーザは、サーバ 20 からアクセスされた項目に対して動作を実行可能である。これらの動作は、内容の編集、サーバから検索された特定の項目の削除、項目の新しい位置への移動、サーバ 20 から検索された項目のコピー、および、その後サーバ 20 上に含むための新規項目の追加を含んでもよい。加えて、この発明の一実施例により、ユーザは、項目に対して実行された動作の組合せを反映するよう、サーバ 20 を更新できるようになる。この発明の一実施例は、ユーザによって実行された動作の可能な組合せを、同等の複合動作へ削減させる。同等の複合動作は、項目を編集し、かつコピーすること、新規項目を作成し、次にそれらを編集および／またはコピーすること、および、既存の項目を編集し、次にそれらを編集することを含んでもよい。

10

【0030】

一実施例では、端末 10 はネットワークインターフェイス 12 を用いてサーバ 20 と通信をやり取りする。一実施例では、ネットワークインターフェイス 12 は、インターネットなどのネットワークを通じた通信を可能とするためのインターネットプロトコル (IP) 通信、および特に伝送制御プロトコル (TCP/IP) を可能とする。また、これに代えて、この発明の実施例は、ローカルエリアネットワーク (LAN)、および他の種類の広域ネットワーク (WAN) などのネットワークを通して、コンピュータ間で通信を伝送してもよい。

【0031】

サーバ 20 は、ファイルシステム 40 を記憶するために、または他の方法で管理するために使用されてもよい。一実施例では、ファイルシステム 40 は複数の部分を含み、各々の部分はユーザまたはアカウントと関連している。ファイルシステム 40 の第 1 の部分 46 は、第 1 の端末 10 へまたは第 1 の端末 10 のユーザへアクセス可能な、サーバ 20 上に記憶されたファイルであってもよい。第 1 の部分 46 は、特定のデータ種類のファイルおよび資源などの複数の項目を含んでもよい。

20

【0032】

第 1 の部分 46 の第 1 の項目 44 が図 1 に識別されている。説明のため、第 1 の項目 44 は文書などの資源であると仮定する。また、これに代えて、第 1 の項目 44 は他の項目を含むファイルであり得る。第 1 の項目 44 は、メタデータ情報および内容を含むかまたは他の方法でそれらと関連している。第 1 の項目 44 のメタデータ情報は、サーバ 20 のメモリ (図示せず) 上の特定の位置 (L1) を識別してもよい。第 1 の項目 44 のメタデータ情報は、サーバ 20 上の第 1 の位置 (L1) を特定するために使用される位置識別情報 (LI1) も含んでもよい。第 1 の項目 44 は資源であると仮定されているため、第 1 の項目 44 はメタデータ情報に関連する内容も含んでいる。

30

【0033】

図示された一実施例では、第 1 の端末 10 は、ネットワークインターフェイス 12 およびネットワーク 15 を通じて伝送された第 1 の通信 32 をサーバ 20 から受取る。第 1 の通信 32 はファイルシステム 40 の第 1 の部分 46 を含む。一実施例では、第 1 の端末 10 を動作させるユーザは、第 1 の部分 46 へのアクセス権を有する。このアクセス権により、ユーザは、第 1 の項目 44 を含む第 1 の部分 46 の一部またはすべてをダウンロードするかまたは他の方法で検索することができる。ユーザは、第 1 の通信 32 を受取った後で、第 1 の部分 46 の作業バージョン 50 を作成できる。作業バージョン 50 は第 1 の部分 46 内の項目からの内容を含む。作業バージョン 50 についてのあるメタデータ情報が、サーバ 20 から転送されて第 1 の通信 32 に含まれてもよい。作業バージョン 50 が作成される際、他のメタデータ情報が第 1 の端末 10 上で作成されてもよい。ファイルシステムから転送されるメタデータ情報は、たとえば、ある項目の位置を特定するためのファイル経路および名前などの位置情報を含んでもよい。

40

【0034】

第 1 の通信 32 で伝送されたデータを用いて、少なくとも第 1 の作業項目 56 を含む作業

50

バージョン 50 を作成してもよい。第 1 の作業項目 56 は、ファイルシステム 40 の第 1 の項目 44 から生じている。一実施例では、第 1 の作業項目 56 の内容部分 58 が第 1 の項目 44 の対応する内容部分 48 からコピーされているために、第 1 の作業項目 56 は第 1 の項目 44 から生じている。

【0035】

ファイルシステム 40 から搬送され得るメタデータ情報は、第 1 の項目 46 の第 1 の位置情報 (L I 1) を含む。第 1 の位置情報 (L I 1) は、第 1 の端末 10 上の第 1 の作業項目 56 についての第 2 の位置 (L 2) を識別するために使用されてもよい。たとえば、第 1 の位置情報 (L I 1) はファイル経路および名前を含んでいてもよい。作業バージョン 50 上にファイル経路を再作成して、第 1 の作業項目 56 が第 2 の位置 (L 2) に位置づけられ得るようにしてもよい。名前も付加的な位置情報として転送されてもよい。多くのアプリケーションでは、名前はファイル経路の包含的部分である。

10

【0036】

作業バージョン 50 が第 1 のコンピュータ 10 上で作成されると、新しいメタデータ情報が記録される。新しいメタデータ情報は、第 1 の作業項目 56 についてのある事象をマークする時刻値を含んでいてもよい。一実施例では、第 1 の時刻値 62 は第 1 の作業項目についての作成時刻に対応してもよい。第 2 の時刻値 64 は第 1 の作業項目 56 についての修正時刻に対応してもよい。第 1 の時刻値 62 および第 2 の時刻値 64 は、作業バージョン 50 が第 1 のコンピュータ 10 上で作成され次第、またはその直後に初期化される。一例として、ユーザがワープロ文書を第 1 の作業項目 56 としてダウンロードする場合がある。文書がダウンロードされると、第 1 の時刻値 62 (作成時刻) および第 2 の時刻値 64 (修正時刻) が、第 1 の端末 10 上のオペレーティングシステム (または他のアプリケーション) によって記録される。たとえば、第 1 の作業項目 56 が作成されると、第 1 の端末 10 は、作成時刻値と修正時刻値とを自動的に記録するウィンドウズ (R) タイプのオペレーティングシステムを実行してもよい。作成時刻とは、特定のコンピュータシステム上で特定の項目に割当てられた、その作成の時刻をマークする値である。作成時刻は、対応する作業項目を、その作業項目が新しいアドレスまたは新しい名前を有していてもその後識別するのに使用可能な静的な値として記憶される。修正時刻とは、作業項目に関連する、項目が編集または作成された最後の時点マークする値である。修正時刻はしたがって、作業バージョン 50 がファイルシステム 40 からダウンロードされた後で変わり得る。

20

30

【0037】

一実施例では、第 1 のコンピュータ 10 は比較ファイル 70 を維持するかまたは他の方法で比較ファイル 70 にアクセスして、メタデータ情報を記憶する。比較ファイル 70 に記憶されたメタデータ情報は、第 1 の作業項目 56 が第 1 のコンピュータ 10 上で作成される際に記録された新しいメタデータ情報と、ファイルシステム 40 の第 1 の項目 44 から搬送されたかもしれないあるメタデータ情報とを含んでいてもよい。

【0038】

提供された例では、比較ファイル 70 は、第 1 の作業項目 56 および第 1 の項目 44 の第 1 の位置情報 (L I 1)、第 1 の作業項目 56 の第 2 の位置 (L 2)、第 1 の作業項目 56 の第 1 の時刻値 62 (作成時刻)、および第 1 の作業項目 56 の第 2 の時刻値 64 (修正時刻) を記憶する。第 1 の位置情報 (L I 1) はサーバから転送され、一方、比較ファイル 70 内の他のメタデータ情報は作業バージョン 50 の作成とともに作成される。作業バージョン 50 内の項目についての最初の時点でのメタデータ情報が作成され、比較ファイル 70 に記憶される。以下により詳細に説明するように、このメタデータ情報は、作業バージョン 50 が作成された後で第 1 の作業項目 56 に対して実行された特定の動作を識別するために使用される。特定の動作を知ることにより、同期化がより効率よく実行され得る。

40

【0039】

比較ファイル 70 が検出に使用され得る動作は、項目の編集、項目の移動、新規項目の作

50

成、項目のコピー、項目の削除、およびそれらの組合せという動作を含む。比較ファイル 70 は、第 1 の通信 32 で伝送された各項目についてのメタデータ情報へのアクセスを提供する。動作が作業バージョン 50 に対して実行された後で、この発明の一実施例は、作業バージョン 50 の項目内のメタデータが比較ファイル 70 に対して比較されることを提供する。作業バージョン項目とファイルシステム項目との違いを判断するために、メタデータ情報の比較が、作業バージョン 50 に対して実行された動作を検出するために用いられる。比較を行なう際、作業バージョン 50 内の項目は、比較ファイル 70 により記録された対応する項目のメタデータとは異なるメタデータ情報を有すると検出されるかもしれない。加えて、作業バージョン 50 内の項目は、比較ファイル 70 により識別された対応する項目を有していないと検出されるかもしれない。比較を行なう際に識別された違いは注記され、作業バージョン 50 をファイルシステム 40 上の第 1 の部分 46 と同期させるために使用される。

10

【0040】

この発明の一実施例の下では、第 1 の部分 46 は、他の端末 25 がサーバ 20 からアクセス可能な共有ファイルである。第 1 の部分 46 は、その一部が第 1 の端末 10 へ伝送された後で別のコンピュータにより変更され得る。他の端末 25 は、たとえば、第 1 の部分 46 内の項目にアクセスし、それに対して動作してもよく、そのため第 1 の部分 46 は、それが第 1 の端末 10 へ伝送された時点から変更される。作業バージョン 50 における変更を識別するための、共有されたファイルシステムとの比較を行なうために、第 2 の通信 34 がサーバ 20 から第 1 の端末 10 へ伝送される。第 2 の通信 34 は、同期化が第 1 の端末 10 で行なわれるべき時点でサーバ 20 上に存在するようなメタデータ情報を含む。一実施例では、同期化要求が第 1 の端末 10 からされ次第、第 2 の通信 34 が第 1 の端末 10 へ伝送される。

20

【0041】

一実施例では、第 1 の端末 10 は同期化動作を実行する。同期化動作は、作業ファイル 50 の変更されたまたは追加された項目と第 1 の部分 46 の項目との間でメタデータ情報を比較してもよい。変更されたまたは追加された作業バージョン項目は、第 1 の端末のユーザが作業バージョン 50 に対して 1 つ以上の動作を実行した結果である。変更されたまたは追加されたファイルシステム項目は、他のユーザがファイルシステム 40 の彼らのバージョンに対して 1 つ以上の動作を実行した結果である。作業バージョン 50 と第 1 の部分 46 の項目との違いが識別され、第 1 の端末 10 のユーザによって調整される。この違いは調整されたメタデータ情報として記録される。一実施例では、調整されたメタデータ情報を第 1 の端末 10 からサーバ 20 へ伝送するために、第 3 の通信 36 が用いられる。調整されたメタデータ情報をサーバ 20 へ伝送して、作業バージョン 50 に対して実行された動作からの変更を反映するようにファイルシステム 40 を更新する 1 つ以上の動作をサーバ 20 に実行させてもよい。さらに、調整された情報がユーザ 50 によって表示され、ファイルシステム 40 を更新するのに使用される変更された項目をユーザが選択できるようにしてもよい。

30

【0042】

別の実施例では、第 1 の部分 46 は他のユーザと共有されておらず、第 1 の端末 10 のユーザのみと共有されている。そのため、第 2 の通信 34 は必要でないかもしれない。むしろ、比較ファイル 70 は、同期化動作を実行し、調整されたメタデータ情報を識別するために使用される。同期化動作が第 1 の端末 10 上で実行された後で、次に調整されたメタデータ情報がサーバ 20 へ伝送される。調整されたメタデータ情報はサーバ 20 へ伝送されて、それに作業バージョン 50 の変更を用いてファイルシステム 40 を更新させるようにする。

40

【0043】

C. 同期化動作

図 2 は、作業バージョン 50 を作成し、その後作業バージョン 50 (図 1) をファイルシステム 40 の対応する部分と同期させるための方法を示す。図 1 の構成要素への言及は、

50

用いられる例示的な構成要素をその実施例と通じさせるよう意図されている。図2で説明されるような一実施例では、第1のファイルシステム40は他のユーザと共有されていないと仮定されている。

【0044】

ステップ210で、ファイルシステム40部分の作業バージョンが第1の端末10上へダウンロードされる。たとえば、第1の端末10はインターネットを通してサーバ20と接続していてもよい。第1の端末10のユーザは、ファイルシステム40の第1の部分46を識別するためのアカウントを有していてもよい。ファイルシステム40の第1の部分46は、第1の端末10上へダウンロードされるよう、ユーザにより選択可能である。

【0045】

ステップ220で、作業バージョン50が作成される際に比較ファイル70が作成される。比較ファイルは作業バージョン50の初期メタデータ情報を記録する。そのメタデータ情報の一部はまた、ファイルシステム40の第1の部分46の項目から転送されてもよい。ステップ210および220は、作業バージョン50に影響を与え得るあらゆる動作に先立って、 $t=0$ で実行される。ステップ230-250はある時刻 $t=i$ たった後で起こり、そのためユーザは作業バージョン50に対する動作を既に実行したかもしれない。その時点で、ユーザは作業バージョン50をファイルシステム40と同期させる要求を行っている。

【0046】

ステップ230で、修正された作業バージョン50と比較ファイル70が作成された時点での作業バージョン50との違いが識別される。この違いはデルタ項目と呼ばれてもよい。デルタ項目は、後の時点での作業バージョン50内の新規項目、コピーされた項目、移動された項目、または修正された項目を含む。デルタ項目は、作業バージョン50に対応していない、比較ファイル70により識別された項目も含んでいてもよい。たとえば、第1の作業項目56は編集および移動を通して動作されるかもしれず、その場合、それは作業バージョン50におけるデルタ項目である。また、これに代えて、比較ファイル70は作業項目56を識別するものの、作業項目56は作業バージョン50から既に削除されているかもしれない。この場合、第1の作業項目56は比較ファイル70におけるデルタ項目である。同様に、比較ファイル70が作成された後で他の作業項目が作業バージョン50へコピーされるかまたは追加されるかもしれず、その場合、これらの項目は作業バージョン50におけるデルタ項目として識別される。

【0047】

ステップ240で、作業バージョン50および比較ファイル70内の項目の違いが識別される。上述のように、これらの違いもデルタ項目と呼ばれる。

【0048】

ステップ250で、作業バージョン50と比較ファイル70により識別された項目との間で識別された違いが調整される。調整のため、ファイルシステム40に更新を指示するためにデルタ項目が選択されてもよい。たとえば、デルタ項目が第1の作業項目56の編集バージョンである場合、その選択は、ファイルシステム40が第1の項目44の編集バージョンを含むべきかオリジナルバージョンを含むべきかを特定する。デルタ項目が作業バージョン50への追加項目（新規項目またはコピーされた項目など）である場合、その選択は、ファイルシステム40がこれらの追加を保持すべきかどうかを判断する。デルタ項目が新しい位置へ移動された第1の作業項目56である場合、その選択は、ファイルシステム40が第1の項目44についての新しい位置情報を使用すべきか、またはファイルシステムが古い位置を維持すべきかを判断する。デルタが作業バージョン50から削除された第1の作業項目56である場合、その選択は、ファイルシステム40が第1の項目44を削除すべきかどうかを特定する。同様の方法が、この出願の他のどこかで詳述されているように、動作の組合せを用いて実行されてもよい。

【0049】

図3は、この発明の一実施例の下で、システム40が他のコンピュータと共有されている

10

20

30

40

50

場合に、作業バージョン50をサーバ20上のファイルシステム40と同期させるための方法を詳述している。図3では、作業バージョン50は図2の一実施例で説明されたように作成されている。ステップ310で、作業バージョン50はファイルシステム40からダウンロードされる。ステップ320で、比較ファイル70が作成されて作業バージョン50およびファイルシステム40についてのメタデータ情報を記録する。ステップ330で、変更が作業バージョン50へなされる。ステップ340で、作業バージョン50と比較ファイル70により識別された項目との違いが識別される。デルタ項目と呼ばれるこれらの項目は、ある時点で1つ以上の動作を介して移動、削除、編集、および追加された作業バージョン項目を含んでいてもよい。

【0050】

ステップ350で、第1の端末10のユーザがサーバ20で同期化要求を行なう。この時点までに、作業バージョン50は1つ以上の動作によりその元々の状態から修正されているかもしれない。

【0051】

ステップ360で、第1の端末10上のファイルシステム40についての新しい情報が受取られる。作業バージョン50が第1の端末10上で作成された時点以降に、ファイルシステム40が他の端末によってアクセスされ、変更されたかもしれない。したがって、ファイルシステム40についての新しい情報は、他のユーザによりファイルシステム40の項目へなされた変更を識別するかもしれない。一実現化例では、ファイルシステム40についての情報はメタデータの形をしており、第1の端末10のユーザによりダウンロードされた第1の部分46の項目に独特のものであってもよい。メタデータ情報は、ダウンロードされた項目に対応するファイルシステム項目の位置情報を含んでいてもよい。加えて、ファイルシステム項目についての新しいメタデータ情報は時刻値を含んでいてもよい。たとえば、ファイルシステム40のデルタ項目を判断するため、同期化が要求された時点でのファイルシステム項目の作成時刻値および修正時刻値が、第1の端末10へ伝送されてもよい。

【0052】

ステップ370で、更新ファイルシステム40と作業バージョンが作成された時点でのファイルシステムとの違い、つまりデルタ項目が検出される。これらのデルタ項目は、ステップ360で受取られた新しいメタデータ情報が、ステップ320で比較ファイルが作成された際に比較ファイル70により識別された項目と比較されることによって識別される。このステップで識別されたデルタ項目は、比較ファイル70によって、またはファイルシステム40について受取られた新しいメタデータ情報によって識別されてもよい。ファイルシステム40についての新しいメタデータ情報によって識別されたデルタ項目は、他のユーザにより移動または編集された項目に対応してもよい。加えて、ファイルシステム40のデルタ項目は、他のユーザにより第1の部分46へ追加された項目を、新規項目または他の項目のコピーとして含んでいてもよい。比較ファイル70によって識別されたデルタ項目は、作業バージョン50が第1のコンピュータ10上で作成された後でファイルシステム40から削除された項目を含む。

【0053】

ステップ380で、ステップ340およびステップ370で識別されたデルタ項目について選択が行なわれる。選択はユーザによって行なわれてもよい。選択は比較ファイル70、作業バージョン50およびファイルシステム40のデルタ項目を特定してもよい。各デルタ項目について、選択はそのデルタ項目を残すべきか否かを判断してもよい。

【0054】

ステップ390で、ステップ340および370で識別された違いの間の矛盾が検出され、解決される。たとえば、作業バージョン50内のある項目が編集される場合があり、そのためそれは、作業バージョン50が作成された時点での対応するファイルシステム項目と比較された場合に、デルタ項目として識別される。比較ファイル70において識別されたそのファイルシステム項目はその後、サーバ20へアクセスを有する別のコンピュータ

10

20

30

40

50

によって変更されるかもしれない。このため、2つのデルタ項目が、比較ファイル70により識別された同じ項目に関連するかもしれない。一実施例では、第1の端末10のユーザは、ファイルシステム40に含めるために2つのデルタ項目のどちらを使用すべきかを選択できる。

【0055】

また、これに代えて、デルタ項目間での矛盾の選択は、各デルタ項目が同期化されたファイルシステム40へ取入れられるべきかどうかを選択する矛盾プロトコルを介して行なわれてもよい。

【0056】

ステップ395で、選択されたデルタ項目を使用してファイルシステム40が更新される。ステップ340および370で識別された各デルタ項目は、ファイルシステムの更新において省略または包含されてもよい。ユーザは矛盾するデルタ項目間で選択できる。

10

【0057】

図4は、この発明の別の実施例の下でサーバ20を動作させるための方法を示す。図4で説明されるような一実施例では、多数のユーザがサーバ20上の共有されるファイルシステムへアクセスすると仮定される。説明のため、図1を参照する。第1の端末10が同期化要求を行なっていると仮定する。ファイルシステム40の一部はサーバ20にアクセス可能な他のクライアント25と共有されている。

【0058】

ステップ410で、共有されるファイルシステム40の一部が、第1の端末10およびクライアント25へ伝送される。各クライアントは別個に動作されて、共有されるファイルシステムの一部へアクセスし、それを受取ってもよい。

20

【0059】

ステップ420で、同期化要求が第1の端末10から受取られる。同期化要求は、ファイルシステム40への変更を実施したいユーザと対応していてもよい。ユーザはまた、ファイルシステム40の一部をダウンロードした他のユーザから入力されたあらゆる変更を受取ることを望んでもよい。

【0060】

ステップ430で、ファイルシステム40についての更新された情報が、同期化を要求しているクライアントへ伝送され得る。ファイルシステム40は、そのクライアントがファイルシステムをダウンロードした時点から更新されて、クライアント25から入力された変更を含んでいるかもしれない。

30

【0061】

ステップ440で、サーバ20は、作業バージョン50に対して実行された動作の結果ファイルシステム40へなすべき変更についての情報を受取る。この変更は、編集、追加（新規項目およびコピー）、削除、および移動などの動作の結果であり得る。

【0062】

ステップ450で、ファイルシステム40は、第1の端末10（同期化要求を行なっているクライアント）から伝送された変更を用いて更新される。更新される変更は、特定のユーザによって、その端末のファイルシステム40の作業バージョンに対する1つ以上の動作の実行後に決定される選択であってもよい。

40

【0063】

ステップ460で、ファイルシステム40へアクセスする他の端末から他に要求が行なわれているか、またはこれから行なわれるかどうかについて判断が下される。同期の要求が他にある場合、要求を行なっている次のクライアントに対してステップ430-460が繰返される。そのような実現化例の下では、共有されるファイルシステム40へのアクセスを有する各クライアントは、それへ付加的な変更を行なう。同期化要求が行なわれると、他のユーザによって行なわれた変更がファイルシステム40に取入れられる。したがってファイルシステム40は、クライアントのうちの1つでの各同期化動作の後で変わり、そのため次のクライアントは以前に更新されたファイルシステム40と同期している。

50

【0064】

D. 作業ファイルに対する動作の検出

図1の例示的な構成要素を参照すると、この発明の実施例は、作業バージョン50の項目とファイルシステム40の項目との同期化を、それらの項目が多数の異なる種類の動作を受けた後でも可能にする。作業バージョン50に対して実行可能な動作は、主要動作または複合動作として特徴付けられてもよい。一実施例の下では、ある項目に対して実行された多数の動作は、1組の同等の複合動作のうちの1つとして検出され得る。

【0065】

一実施例では、主要機能は編集、削除、コピー、移動および新規作成である。編集動作は、作業バージョン50内の項目の内容が変更されることをもたらす。削除動作は、項目が作業バージョン50から除去されるようにする。コピー動作は、作業バージョン50内の項目の内容を、新規項目または追加項目として再作成する。移動動作は、作業バージョン50のある位置に位置する項目が、新しい位置を与えられるようにする。位置は、名前、メモリアドレスおよび記憶装置によって規定されてもよい。このように、移動動作は、項目を新しいフォルダ位置へ移動させるため、項目の名前を変えるため、または項目を新しい記憶装置へ移動させるために実行されてもよい。新規作成動作は、付加的な項目を作業バージョン50へ作成するために作業バージョン50上で実行される。

【0066】

複合動作は、作業バージョン項目を作成するため、および／または作業バージョン項目に影響を与えるために、作業バージョン50上で実行される多数の動作の組合せである。この発明の実施例とは対照的に、以前の同期化システムは、主要動作のいくつかの実行を検知できるが、ある主要動作または動作の組合せを検知できない。この発明の実施例によって提供される利点により、作業バージョン50の個々の項目に対して実行されたすべての主要動作、および多数の動作の組合せの検出および同期化が可能になる。

【0067】

ファイル管理の動作を説明するための分析的表現が、フォーマットaObを用いて説明されてもよい。ここで、大文字は動作を表わし、動作の前の項目は動作についての源を表わし、動作の後の項目は動作についての宛先を表わしている。主要動作を要約すると、以下のようになる。

【0068】

E x — ファイルXを編集
D x — ファイルXを削除
N x — 新規項目Xを作成
x M y — XをYへ移動
x C y — XをYとしてコピー

この発明の一実施例の下では、複合動作は、有限数の同等の複合動作へ削減および要約可能である。これらの要約を作成する際に用いられる原則のいくつかの例は、以下を含む。

(1) 項目が削除されている場合、その項目に対して実行された以前の動作は無視され得る。(2) ある項目の多数の移動は、最初の源から最後の宛先への1回の移動として扱われ得る。(3) 動作の組合せに対して実行されたあらゆる移動動作は、他の動作に関して任意の順序で分析可能であり、そのため、移動が別の動作の前に実行されたと仮定することが、真の、かつ簡略化された結果を提供する。これらの原則を用いると、どの作業項目も9つの可能な動作または動作の組合せのうちの1つを受けると仮定することができ、ここで、動作の組合せは他の動作組合せの同等物である。作業バージョン50の項目に対して実行される動作は、5つの主要動作のうちの1つとして、または4つの同等の複合動作のうちの1つとして、ファイルシステム40のために再現可能である。一実施例では、4つの同等な動作の組合せは以下のとおりである。

【0069】

E x M y — Xを編集し、それをYへ移動
(N x) C y — Xを作成し、それをYとしてコピー

E ((N x) C y) ← Xを作成し、それをYとしてコピーし、Yを編集

E (x C y) ← XをYとしてコピーし、Yを編集

挿入句は、どの同等の複合動作においても、最初に行われるべきである。

【0070】

図5は、この発明の一実施例の下で、ユーザが作業バージョン50をファイルシステム40と同期させるよう要求した時点で作業バージョン50内の項目について実行されている動作を検出するための方法を示す。図示されたような一実施例では、同期化の時点で作業バージョン50内の各項目につき10の可能な結果がある。つまり、変更なし、5つの主要動作、および4つの動作の同等の組合せである。

【0071】

ステップ502で、ファイルシステム40の一部から作業バージョン50が作成される。ステップ504で、作業バージョン50についての情報を含む比較ファイル70が作成される。ステップ502および504は双方とも、任意の動作が作業バージョン項目について実行される前に（つまり $t = 0$ で）起こると仮定される。後の時点で（つまり $t = f$ で）、ユーザがファイルシステムとの同期化を要求する。 $t = 0$ と $t = f$ との間から、ユーザは作業バージョン50を変更する1つ以上の動作を実行してもよい。

【0072】

同期化要求を受取ると、ステップ506で、比較ファイル70により識別され位置が特定された項目が作業バージョン50内の対応する項目と同じ位置を有しているかどうかについて、判断が行われる。最初に比較ファイル70が作られた際に、各項目の位置が記録される。このため、ステップ506は、比較ファイル70において識別された項目の位置がまだ特定可能かどうかを、その項目について最初に記録された位置情報を用いて判断する。

【0073】

ステップ506が、比較ファイル70により識別された項目はまだ作業バージョン50内の同じ位置を有すると判断した場合、ステップ508が、その項目が比較ファイル70に記録された後で編集されたかどうかに関する別の判断とともに続く。ステップ508がその項目は編集されなかったと判断した場合、次にステップ510が、その特定の項目は作業バージョン50内で変更されなかったと結論付ける。ステップ508が特定の項目は編集されたと判断した場合、次にステップ512が、比較ファイル70により識別された項目を、編集されたとして注記する。

【0074】

ステップ506が、比較ファイル70により識別された項目がその項目について記録された情報によって位置を特定されなかったと判断した場合、次にステップ514が、その項目が移動されたかどうかについての判断を行なう。判断がその項目は移動されなかったというものであれば、ステップ516がその項目を、削除されているとして注記する。判断がその項目は移動されたというものであれば、次にステップ518が作業バージョン50内の項目の新しい位置を注記する。次にステップ520で、移動された項目が編集もされたかどうかについての判断が行われる。判断が肯定的であれば、ステップ522で項目は、移動され、かつ編集されたとしてマークされる。

【0075】

ステップ524は、比較ファイル70により識別された、チェックされるべき項目が他に残っているかどうかについての判断を行なう。前の項目が変更なしと判断された場合、ステップ524はステップ510に続く。前の項目が編集されているとして注記された場合、ステップ524はステップ512に続く。前の項目が削除されたとして注記された場合、ステップ524はステップ516に続く。前の項目が移動されたとして注記された場合、ステップ524はステップ520に続く。前の項目が移動され、かつ編集されたとして注記された場合、ステップ524はステップ522に続く。ステップ524が、比較ファイル70により識別された項目が残っており、これらの項目はまだチェックされていないと判断した場合、次にステップ526が、比較ファイル70により識別された、チェック

10

20

30

40

50

されるべき次の項目を提供する。次の項目について、この方法がステップ506から始まって繰返される。

【0076】

ステップ524が、比較ファイル70により識別された、チェックされるべき項目が残っていないと判断した場合、ステップ528が、作業バージョン50に未チェックのまま残っている項目があるかどうか判断する。未チェックの項目とは、ステップ506-524で比較ファイル項目がチェックされた結果、チェックされなかった作業バージョン項目である。作業バージョン50に未チェックの項目がない場合、この方法は終了する。その他の場合、ステップ530で、作業バージョン50内に残った項目がチェック済とマークされる。

10

【0077】

ステップ532は、作業バージョン50内の未チェック項目がコピーであるかどうかを判断する。判断が肯定的である場合、ステップ534はその項目をコピーとして注記する。ステップ536は、コピーされた項目が作業バージョン50内で新規作成された別の項目からコピーされたかどうかを判断する。

【0078】

ステップ536での判断が否定的である場合、ステップ538が、コピーされた項目が作成後に編集もされたかどうかをステップ538で判断する。ステップ538での判断が肯定的である場合、ステップ540が、コピーされ、かつ編集されたと注記する。

【0079】

20

ステップ536での判断が肯定的である場合、ステップ542で、その項目は新規かつコピーされたとしてマークされる。言換えれば、その項目は、作業バージョン50がファイルシステム40から作成された後で、オリジナルとして作成された別の項目からコピーされているとして注記される。ステップ544で、新規かつコピーされた項目が編集もされたかどうかについて判断がなされる。ステップ544での判断が肯定的である場合、その項目は、新規でコピーされかつ編集されたとして注記される。

【0080】

ステップ532で判断がその項目はコピーではないというものである場合、次にステップ548でその項目は新規であるとして注記される。

【0081】

30

ステップ554は、未チェックとしてマークされた項目が作業バージョン50内に残っているかどうかを判断する。ステップ544は、ステップ538での判断が否定的で項目はコピーされているとしてのみ注記される場合；項目はコピーされかつ編集されていると判断された場合にはステップ540；項目は新規のみであると判断された場合にはステップ548；ステップ544での判断が否定的で項目は新規かつコピーされていると判断された場合；および、項目は新規で編集されかつコピーされていると判断された場合にはステップ546というこれらのステップのうちの1つに続く。ステップ554が作業バージョン50に未チェック項目が残っていると判断した場合、ステップ556が次の未チェック項目へ反復する。それからこの方法が次の項目について、ステップ532から始まって繰返される。ステップ554が作業バージョン50に未チェック項目が残っていないと判断した場合、この方法は終了する。

40

【0082】

図5の実施例によって示すように、同期化動作は、同期化される各項目について10の可能な結果を検出し得る。各項目は、ダウンロード以降変更なしと判断されるかもしれない（ステップ510）。その他の場合、同期化される各項目は、作業バージョン50が作成された後でユーザにより実行された1つ以上の動作の結果であると判断され得る。5つの主要動作、つまり、編集（ステップ512）、移動（ステップ518）、削除（ステップ516）、新規作成（ステップ548）、およびコピー作成（ステップ534）が検出される。加えて、4つの複合動作、つまり、移動かつ編集（ステップ522）、新規かつコピー（ステップ542）、新規、編集かつコピー（ステップ546）、およびコピーかつ

50

編集（ステップ540）が検出される。

【0083】

一実施例では、作業バージョン項目が削除され、次に同じ名前および位置情報で再作成される特別の場合が起こり得る。そのような項目は、ステップ506がその特別な場合用にチェックを導入していれば、移動された項目というよりもむしろ新規項目として識別され得る。具体的には、再作成された項目の作成時刻値などの識別情報を用いて、その項目がステップ506において削除および再作成の対象とならなかったことをチェックしてもよい。

【0084】

図6-9は、この発明の実施例の下で、図5において実行された動作を検出するためのさらなる詳細を提供するフローチャートを示す。図6は、ある項目が移動または削除されたかどうかを判断するための方法である。この発明の一実施例の下では、図6は図5のステップ514-518に対応していてもよい。

10

【0085】

ステップ610では、作業バージョン50が第1の端末10上に作成される際、作業バージョン50内の各項目について第1の時刻値が記録される。一実施例では、第1の時刻値は項目の作成時刻に対応していてもよい。作成時刻は、ウィンドウズ（R）などのあるオペレーティングシステムの下で項目に付加される特性である。作成時刻は、第1の端末10上で作成された項目についての時刻値を、その項目が別のコンピュータからダウンロードされる際に記録してもよい。このため、作業バージョン50の項目がファイルシステム40からダウンロードされる際に、第1の端末10はその項目の作成時刻を記録してもよい。作成時刻は、1秒の1000分の1まで、または数倍大きい規模まで有意であってもよい。

20

【0086】

ステップ620で、作業バージョン50の作成時に、各項目について位置情報が作成される。位置情報は、ファイルシステム40上または作業バージョン50上の項目の位置を特定するために使用可能なファイル経路または名前のセグメントに対応していてもよい。ステップ610および620の双方は、作業バージョン50が作成されるときに対応して、および任意の動作が実行される前に、 $t = 0$ で起こる。第1の時刻値および初期位置情報は、比較ファイル70に位置してもよい。

30

【0087】

ステップ620の後、同期化要求がなされたときまたは $t = f$ で、フローチャートは前進する。ステップ630は、最初に（ $t = 0$ で）記録された位置情報が $t = f$ で作業バージョン50内の項目の位置を特定するかどうかを判断する。ステップ630での判断が肯定的である場合、次にステップ640がその項目を移動されていないとして記録する。判断が否定的である場合、ステップ650が続く。ステップ650は、 $t = f$ での作業バージョン50内の任意の項目が、ステップ610で位置が未特定の項目について記録された時刻値と整合する、対応する第1の時刻値を有するかどうかを判断する。一実施例では、位置が未特定の項目の作成時刻に整合する作成時刻について、作業バージョン50内の他の項目がチェックされてもよい。

40

【0088】

作成時刻が1秒の1000分の1またはさらに100万分の1を超えて実行され得ることを仮定すると、行方不明の項目と同じ作成時刻を有する作業バージョン50内の別の項目は、新しい位置にある、位置が未特定の項目であると仮定することができる。ステップ650での判断が肯定的である場合、ステップ660で、同じ時刻値を有する項目が移動されているとして記録される。ステップ650が、行方不明の項目の作成時刻を有する項目が作業バージョン50内にないと判断した場合、ステップ670がその項目を、削除されているとして注記する。

【0089】

図7は、この発明の一実施例の下で、作業バージョン50内の任意の項目が編集動作の対

50

象となったかどうかを判断するためのフローチャートである。図7により図示される方法は、図5のステップ508、510および512に対応していてもよい。ステップ710で、作業バージョン50において第1の時刻値が識別される。時刻値はその項目の作成時刻に対応していてもよい。ステップ720で、第2の時刻値が同じ項目について、その項目の修正時刻に対応して識別される。前述のように、作成時刻および修正時刻は双方とも、ウィンドウズ(R)などのオペレーティングシステムによって自動的に記録される時刻値である。これらの時刻値は双方とも、1秒の1000分の1まで、またはさらにはるかに大きな精度の桁数まで有意であってもよい。このため、図7によって示すような実施例については、作成時刻および修正時刻は双方とも、その項目に固有のものであると仮定される。

10

【0090】

ステップ730で、修正時刻が作成時刻と異なるかどうかについての判断がなされる。項目がオリジナル、コピーまたはダウンロードのいずれかとして作成されると、一実施例は、作成時刻と修正時刻とは同じであると規定する。このため、作成時刻と修正時刻とが異なっている場合、ステップ740がその項目を編集されたとして注記する。その他の場合、ステップ750がその項目を編集されていないとして注記する。

【0091】

作成時刻と修正時刻とが当初、必ずしも全く同じではないものの、互いの範囲内にあるということは可能である。一実施例は、修正時刻が作成時刻の範囲外にあるかどうかを判断するようチェックしてもよい。

20

【0092】

図8は、この発明の一実施例の下で、編集かつ移動という複合動作を識別するための方法である。一実施例では、図8によって示すような方法が、図5のステップ514、518、520および522のサブステップとして使用されてもよい。

【0093】

ステップ810で、作業バージョンがファイルシステム40からダウンロードされる際、作業バージョン50内の各項目について多数の時刻値が記録される。他の実施例において注記されているように、記録された時刻値のうちの第1のものは作成時刻に対応する。作成時刻は、作業バージョン50を使用しているコンピュータのオペレーティングシステムによって自動的に作成されてもよい。作成時刻および修正時刻は各々、比較ファイル70に記録され、対応する項目に関連付けられてもよい。

30

【0094】

ステップ820で、作業バージョン50が作成される際に各項目について位置情報が記録される。位置情報は、作業バージョン50内の項目の位置を特定し得るファイル経路のセグメントを含んでいてもよい。位置情報は項目の名前も含んでいてもよい。作業バージョン内の各項目についての初期位置情報が比較ファイル70に記録されてもよい。

【0095】

ステップ830で、最初に記録された位置情報がその後、作業バージョン50内の対応する項目の位置を特定できるかどうかについて判断がなされる。位置情報が対応する項目の位置を特定する場合、ステップ840で、その項目は移動されていないとして記録される。最初に記録された位置情報が対応する項目の位置を特定しない場合には、ステップ850において別の判断がなされる。ステップ850は、作業バージョン50内の別の項目が、位置が未特定の項目と同じ作成時刻を有するかどうかを判断する。この判断が否定的である場合には、ステップ860が、位置が未特定の項目を削除されたとして注記する。

40

【0096】

その他の場合、ステップ870が、修正時刻がその項目についての作成時刻と整合するかどうかについての判断を行なう。ステップ870での判断が肯定的である場合、ステップ880で、その項目は移動のみされているとして注記される。ステップ870での判断が否定的である場合、ステップ890で、その項目は移動かつ編集されているとして注記される。

50

【0097】

図9は、この発明の一実施例の下で、作業バージョン50内の未チェック項目に対する1つ以上の動作を検出するためのプロセスを示す。図9によって示すプロセスで、未チェック項目に対して実行される動作は、新規作成、コピー、および編集からなる群からの少なくとも2つの動作を含んでいてもよい。一実施例では、図9によって示すようなプロセスが、図5のステップ532-546のサブステップを形成してもよい。

【0098】

ステップ910-980は、同期化が要求された時点で、作業バージョン50内の個々の未チェック項目に対して実行される。ステップ910-980は、作業バージョン50上で実行されたかもしれない他の動作を検出するためにある他のステップが既に実行された、ということを仮定している。具体的には、910-980は作業バージョン50内の未チェック項目に対して実行される。図5に示すように、未チェック項目とは、比較ファイル70によって識別された項目が作業バージョン50の項目と比較された後で残された項目である。未チェック項目はしたがって、作業バージョン50が作成された後で作成されたと仮定され得る。このため、未チェック項目はコピーおよび/または新規項目である。未チェック項目は、作成された後で編集もされているかもしれない。

【0099】

ステップ910で、作業バージョン内の各未チェック項目について時刻値が記録される。作業バージョン50がダウンロードされた後で作成された項目については、作成時刻は、ユーザがその項目を作成し、それをダウンロードされた項目とともに作業バージョン50に記憶させた時点に対応していてもよい。作成時刻は、作業バージョン50内の各未チェック項目に、固有の識別子を提供すべきである。加えて、作業バージョン50内の各項目についての修正時刻が記録される。修正時刻は、対応する項目が編集されるたびに変わる。しかしながら、項目が編集されない場合、修正時刻は、その同じ項目についての作成時刻と同じであるかまたは非常に近いものであるべきである。一実施例では、作業バージョン50内の各項目についての作成時刻が、その項目について比較ファイル70に記憶された作成時刻と整合する、ということが仮定できる。

【0100】

ステップ920は、各未チェック項目についての修正時刻が比較ファイル70に記憶された項目のうちの1つの修正時刻と整合するかどうかを判断する。一実施例は、コピーの修正時刻はそのオリジナルの修正時刻と同じであると規定する。この特徴は、第1の端末10上で動作されるアプリケーションを通して実現されてもよい。一実施例では、第1の端末10は、この属性または特徴を含むオペレーティングシステムを動作させる。そのようなオペレーティングシステムの一例はウィンドウズ(R)タイプのオペレーティングシステムである。

【0101】

判断が肯定的である場合、ステップ930が、その項目はもともとファイルシステム40からダウンロードされた別の項目のコピーとして作成されているとして注記されると規定する。編集動作が修正時刻を変更しているため、その項目はその後編集されなかったと仮定され得る。判断が否定的である場合には、ステップ940が続く。

【0102】

ステップ940は、未チェック項目の修正時刻が作成時刻の前であるかどうかを判断する。修正時刻が作成時刻の後である場合、ステップ950が、その項目を別の項目のコピーとして注記する。これは、別の項目のコピーはオリジナルの修正時刻を保持するものの、作成時に新しい作成時刻を割当てられるためである。ステップ940は、コピーとして作成された項目がその後編集されたかどうかを判断するのには使用できない。なぜなら、それは修正時刻を作成時刻の後になるよう変更するからである。

【0103】

修正時刻が作成時刻の後である場合、ステップ960が、修正時刻が作成時刻と整合するかどうかについての判断を行なう。項目が新規としてまたは別の項目のコピーとして作成

10

20

30

40

50

されている場合、修正時刻と作成時刻とは全く同じであるか、またはオペレーティングシステムの構成もしくは作業バージョン50に影響を与える他のアプリケーションに依存して若干異なるかもしれない。ステップ960からの判断が肯定的である場合、ステップ970が、その項目は新規項目を作成する動作の結果として作成されていると規定する。

【0104】

ステップ960からの判断が否定的である場合、ステップ980が、その項目は編集され、新規でかつおそらくコピーでもあると規定する。このため、ステップ980は2つの可能性を提供する。この時点で、修正時刻と作成時刻とは、2つの可能性を区別するためには使用できない。2つの可能性間で解決するため、一実施例が、ステップ980で識別された全項目も内容整合の1つ以上のステップの対象となると規定してもよい。ファイルが新規でかつ編集されているか、または新規でコピーされかつ編集されているかを判断するために、アルゴリズムを用いて、ステップ980における全ファイルの内容を、作業バージョン50内の新規であるとして識別された他のファイルの内容と比較してもよい。その前提は、後者が、新規であるとして識別された別の項目と同様の内容を有するであろうということであってもよい。

【0105】

E. ユーザインターフェイス

図10は、この発明の一実施例で使用されるユーザインターフェイス1000を示す。ユーザインターフェイス1000により、ユーザは、異なるコンピュータ上で変更された同じ項目間で選択できるようになる。たとえば、図1を参照すると、第1の端末10上のユーザは、ファイルシステム40からダウンロードされた作業バージョン項目に対して動作を実行してもよい。ファイルシステム40は共有されてもよい。他のユーザがネットワークを通してそれにアクセスするかもしれない。他のユーザがファイルシステム40内のある項目に対して動作を行なう一方、第1の端末10上のユーザがその対応する作業バージョン項目に対して動作を行なうかもしれない。同期化要求が行なわれると、矛盾が提示されるかもしれない。作業バージョンの変更された項目に対応するファイルシステム項目は、ファイルシステム40にアクセスした別のユーザによってすでに変更されている。

【0106】

この発明の一実施例により、同期化要求を行なうユーザは、ファイルシステム40内の項目と対応する作業バージョン50内の項目との間で選択できるようになる。また、一実施例により、要求を行なっているユーザは、他のコンピュータにより更新されたファイルシステム項目と、同期化要求を行なっているコンピュータ上の作業バージョン項目との間の矛盾をいかに解決するかを選択できるようになる。

【0107】

ユーザインターフェイス1000は、第1の列1110と第2の列1120とを含む。第1の列は、第1のコンピュータ10上のデルタ項目についての情報を提供する。第2の列1120は、ファイルシステム40のデルタ項目についての情報を提供する。ファイルシステム40のデルタ項目は、更新されたファイルシステムを比較ファイル70と比較することによって識別されてもよい。第1の列1110の第1の部分1125は、作業バージョン50のデルタ項目を識別する。第2の列1120の第1の部分は、他のユーザによって更新されたような、ファイルシステム40のデルタ項目を識別する。第1の列1110の第2のセグメント1118は、作業バージョン50のデルタ項目に対して実行された動作または同等の複合動作を識別する。同様に、第2の列1120の第2のセグメント1128は、更新されたファイルシステムのデルタ項目に対して実行された動作または同等の複合動作を識別する。第2のセグメント1128に列挙された動作は、共有されるファイルシステム40へアクセスする他者により実行されたと仮定される。

【0108】

第1の列1110および第2の列1120に列挙された各デルタ項目について、ユーザは、変更を保持するかまたは項目をそのままファイルシステム40上に維持するかを選択してもよい。第1の列1110に列挙されたデルタ項目が第2の列1120のデルタ項目と矛

盾する場合、ユーザはその矛盾をいかに解決するかを決定してもよい。たとえば、ファイルシステム40内のある項目が作業バージョン50上へダウンロードされ、その後作業バージョン50内で動作される場合がある。ダウンロードされた同じ項目が別のコンピュータによってアクセスされ、異なる態様で動作されるかもしれない。同期化要求がなされると、その要求をしたコンピュータは矛盾を提示される。そのコンピュータのユーザはその矛盾を解決する能力を与えられてもよい。そのユーザはどのデルタ項目を保持すべきかを選択し、ファイルシステム40との調整時にその項目を使用することができる。

【0109】

F. ハードウェアの説明

図11は、この発明の一実施例が実現され得るコンピュータシステム1100を示すブロック図である。コンピュータシステム1100は、情報を通信するためのバス1102または他の通信機構と、情報を処理するためにバス1102と結合されたプロセッサ1104を含む。コンピュータシステム1100は、プロセッサ1104により実行されるべき命令および情報を記憶するためにバス1102に結合された、ランダムアクセスメモリ(RAM)または他のダイナミック記憶装置などのメインメモリ1106も含む。メインメモリ1106は、プロセッサ1104により実行されるべき命令の実行中に、一時的な変数または他の中間情報を記憶するためにも使用されてもよい。コンピュータシステム1100はさらに、プロセッサ1104用の命令およびスタティック情報を記憶するためにバス1102に結合された読出専用メモリ(ROM)1108または他のスタティック記憶装置を含む。磁気ディスクまたは光ディスクなどの記憶装置1110が、情報および命令を記憶するために提供され、バス1102へ結合されている。

【0110】

コンピュータシステム1100は、バス1102を介して、情報をコンピュータユーザへ表示するための陰極線管(CRT)などのディスプレイ1112へ結合されてもよい。英数字キーおよび他のキーを含む入力装置1114が、情報およびコマンド選択をプロセッサ1104へ通信するためにバス1102へ結合されている。ユーザ入力装置の別の種類はマウス、トラックボールまたはカーソル方向キーなどのカーソル制御1116であり、方向情報およびコマンド選択をプロセッサ1104へ通信し、ディスプレイ1112上のカーソルの動きを制御する。この入力装置は通常、2つの軸、つまり第1の軸(たとえばx)および第2の軸(たとえばy)において2つの自由度を有しており、それによりこの装置は平面における場所を特定することができる。

【0111】

この発明は、ここに記載された手法を実現するためのコンピュータシステム1100の使用に関する。この発明の一実施例によれば、これらの手法は、プロセッサ1104がメインメモリ1106に含まれた1つ以上の命令の1つ以上の連続を実行するのに応答して、コンピュータシステム1100により実施される。そのような命令は、記憶装置1110などの別のコンピュータ可読媒体からメインメモリ1106へ読込まれてもよい。メインメモリ1106に含まれる命令の連続の実行により、プロセッサ1104はここに記載されたプロセスステップを行なうようになる。代替的な実施例では、この発明を実施するために、ソフトウェア命令の代わりにまたはそれと組合せて、配線回路が使用されてもよい。このため、この発明の実施例は、配線回路とソフトウェアのどの特定の組合せにも限定されない。

【0112】

ここに使用されるような用語「コンピュータ可読媒体」は、実行用にプロセッサ1104へ命令を提供することに関与するあらゆる媒体を指す。そのような媒体は、不揮発性媒体、揮発性媒体および通信媒体を含むもののそれらに限定されない多くの形態をとってもよい。不揮発性媒体は、たとえば、記憶装置1110などの光ディスクまたは磁気ディスクを含む。揮発性媒体はメインメモリ1106などのダイナミックメモリを含む。通信媒体は、バス1102を構成する配線を含む同軸ケーブル、銅線および光ファイバを含む。通信媒体は、電波および赤外線データ通信中に発生するものなどの音波または光波の形もと

10

20

30

40

50

り得る。

【0113】

コンピュータ可読媒体の通常の形態は、たとえば、フロッピー（Ｒ）ディスク、フレキシブルディスク、ハードディスク、磁気テープ、または他の磁気媒体、ＣＤ－ＲＯＭ（コンパクトディスクＲＯＭ）、他の光媒体、パンチカード、紙テープ、孔のパターンを有する他の物理的媒体、ＲＡＭ、ＰＲＯＭ（プログラマブルＲＯＭ）、ＥＰＲＯＭ（消去可能プログラマブルＲＯＭ）、フラッシュＥＰＲＯＭ、他のメモリチップまたはカートリッジ、以下に記載されるような搬送波、またはそこからコンピュータが読出可能なあらゆる媒体を含む。

【0114】

コンピュータ可読媒体のさまざまな形態は、実行用にプロセッサ 1104 へ 1 つ以上の命令の 1 つ以上の連続を搬送することに関与してもよい。たとえば、命令はまず遠隔コンピュータの磁気ディスクに載って搬送されてもよい。遠隔コンピュータは命令をそのダイナミックメモリへロードし、モデムを用いた電話回線を通して命令を送ることができる。コンピュータシステム 1100 にとって局所的なモデムは、電話回線上のデータを受取り、赤外線送信機を用いてデータを赤外線信号に変換することができる。赤外線検出器は赤外線信号において搬送されたデータを受取可能であり、適切な回路がそのデータをバス 1102 上へ置くことができる。バス 1102 はデータをメインメモリ 1106 へ搬送し、そこからプロセッサ 1104 が命令を検索して実行する。メインメモリ 1106 によって受取られた命令は、プロセッサ 1104 による実行の前または後のいずれかで、随意に記憶装置 1110 上に記憶されてもよい。

【0115】

コンピュータシステム 1110 は、バス 1102 に結合された通信インターフェイス 1118 も含む。通信インターフェイス 1118 は、ローカルネットワーク 1122 へ接続されたネットワークリンク 1120 へ双方向データ通信結合を提供する。たとえば、通信インターフェイス 1118 は、データ通信接続を対応する種類の電話回線へ提供するデジタル総合サービス網（ＩＳＤＮ）カードまたはモデムであってもよい。別の例として、通信インターフェイス 1118 は、データ通信接続を互換性がある ＬＡＮ へ提供するローカルエリアネットワーク（ＬＡＮ）カードであってもよい。無線リンクも実現されてもよい。そのような実現化例では、通信インターフェイス 1118 は、さまざまな種類の情報を表わすデジタルデータストリームを搬送する電気、電磁または光信号を送り、受取る。

【0116】

ネットワークリンク 1120 は通常、1 つ以上のネットワークを通して他のデータ装置へデータ通信を提供する。たとえば、ネットワークリンク 1120 は、ホストコンピュータ 1124 へ、またはインターネットサービスプロバイダ（ＩＳＰ） 1126 により動作されるデータ装置へ、ローカルネットワーク 1122 を通して接続を提供してもよい。ＩＳＰ 1126 は次に、現在では一般に「インターネット」 1128 と呼ばれる全世界的パケットデータ通信ネットワークを通して、データ通信サービスを提供する。ローカルネットワーク 1122 およびインターネット 1128 は双方とも、デジタルデータストリームを搬送する電気、電磁または光信号を使用する。コンピュータシステム 1100 へ、またはコンピュータシステム 1100 からデジタルデータを搬送する、さまざまなネットワークを通る信号、および、ネットワークリンク 1120 上の、ならびに通信インターフェイス 1118 を通る信号は、情報を伝達する搬送波の例示的な形態である。

【0117】

コンピュータシステム 1100 は、ネットワーク、ネットワークリンク 1120 および通信インターフェイス 1118 を通して、メッセージを送り、プログラムコードを含むデータを受取る。インターネットの例では、サーバ 1130 は、アプリケーションプログラムについての要求されたコードを、インターネット 1128、ＩＳＰ 1126、ローカルネットワーク 1122、および通信インターフェイス 1118 を通して送信してもよい。

【0118】

10

20

30

40

50

受取られたコードは、それが受取られたときにプロセッサ 1 1 0 4 によって実行されてもよく、および／または後の実行のために記憶装置 1 1 1 0 または他の不揮発性記憶装置に記憶されてもよい。このように、コンピュータシステム 1 1 0 0 は、搬送波の形をしたアプリケーションコードを獲得し得る。

【0 1 1 9】

G. 代替的な実施例

ここに提供された実施例（たとえば図 1 を参照）は調整情報をメタデータ情報の形態であるとして説明しているが、他の実施例は、第 1 の部分 4 6（図 1）内の項目についての内容の一部またはすべてを用いて、作業バージョン 5 0（図 1）の項目への変更を識別してもよい。特に、ある項目が別のものからコピーされたかどうかを判断するために内容整合が用いられてもよい。知的なアルゴリズムを採用し、特定の種類の類似性を有する項目は互いのコピーであるという仮定を用いて、項目の内容間の類似性を検出してもよい。

10

【0 1 2 0】

内容整合は、図 9 で説明されたようなプロセスでのさらなるステップとしても用いられてもよい。たとえば、ステップ 9 7 0 によって示すように同等の動作が検出された場合、その項目が新規で、編集されかつコピーもされているかどうかを判断することができないかもしれない。ある項目が、新規項目であった別の項目の編集されたコピーであるかどうかを判断するために、内容整合が必要とされるかもしれない。

【0 1 2 1】

内容整合の別の使用は、項目の一方または双方の時刻値が別の項目と全く同じである場合に、決着をつけるものとしてである。ウィンドウズ（R）などの一般的なオペレーティングシステムに適用された時刻値の著しい桁（つまり 1 秒の 1 0 0 万分の 1）を考慮すると、2 つの項目が全く同じ作成時刻または修正時刻を有する可能性はわずかしかない。しかしながら、異なる項目の時刻値間に完全な整合がある場合、この発明の実施例は、2 つの項目間で解説するために内容整合を可能とする。

20

【0 1 2 2】

この発明の実施例は異なるコンピュータ上で動作されるファイルの同期化について説明されてきたが、他の実施例がスタンドアロンまたは単一のコンピュータシステムへ適用され得ることが留意されるべきである。たとえば、この発明の一実施例についてのある用途は、多数の入力を含む 1 つのファイルを、アーカイブとして作成されたバックアップファイルと同期させることである。他のコンピュータシステムとの対話は必要とされないであろう。

30

【0 1 2 3】

いくつかの用途では、ある同等の複合動作を検出せず、むしろより簡単な動作がある項目に対して行なわれたと仮定することがより有用であるかもしれない。また、これに代えて、同等の複合動作が検出されるものの、他の動作を用いてファイルシステム 4 0 を更新してもよい。たとえば、この発明の一実施例は、 $(N x) C y$ という同等の複合動作を、 $N x$ かつ $N y$ として扱ってもよい。このため、同期化の最中、ファイルシステム 4 0 は 2 つの新規項目を追加するよう命令されるであろう。同様に、 $E (x C y)$ という複合動作は $N y$ として扱われてもよく、この場合ファイルシステム 4 0 は、 X を Y へコピーして次にそれを編集するというよりもむしろ、1 つの新しいファイルを作成するよう命令されるであろう。

40

【0 1 2 4】

H. 結論

前述の明細書で、この発明がその特定の実施例を参照して説明されてきた。しかしながら、さまざまな修正および変更がそれに対して、この発明のより幅広い精神および範囲から逸脱することなくなされることは明らかである。したがって、明細書および図面は限定的な意味というよりもむしろ例示的な意味において考えられるべきである。

【図面の簡単な説明】

【0 1 2 5】

50

【図 1】この発明の一実施例の下でのシステムアーキテクチャの概要を示す図である。

【図 2】この発明の一実施例の下での、端末上での同期化を説明するフローチャートである。

【図 3】この発明の一実施例の下での、システムの端末上での、多くのユーザにより共有されているファイルの同期化を説明するフローチャートである。

【図 4】この発明の一実施例の下での、多数の端末上での共有されたファイルシステムの同期化を詳述するフローチャートである。

【図 5】この発明の一実施例の下での、ファイルシステムの作業バージョンに対する多数の動作および複合動作の検出を詳述するフローチャートである。

【図 6】同期化動作中に移動または削除された項目を識別するためのフローチャートである。 10

【図 7】この発明の一実施例の下での、同期化動作中に編集された項目を識別するためのフローチャートである。

【図 8】この発明の一実施例の下での、編集され、および／または移動され、もしくは削除された項目についての 1 つ以上の動作を識別するためのフローチャートである。

【図 9】この発明の一実施例の下での、新規または他の項目からのコピーとして作成され、かつおそらく編集された項目を識別するためのフローチャートである。

【図 10】この発明の一実施例で使用するユーザインターフェイスを示す図である。

【図 11】この発明の一実施例で使用するハードウェアブロック図である。

【図 1】

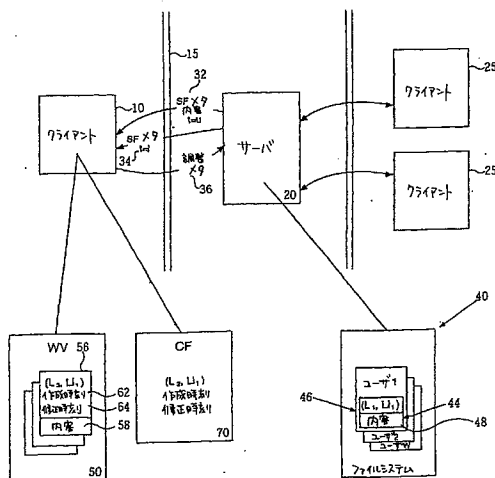


FIGURE 1

【図 2】

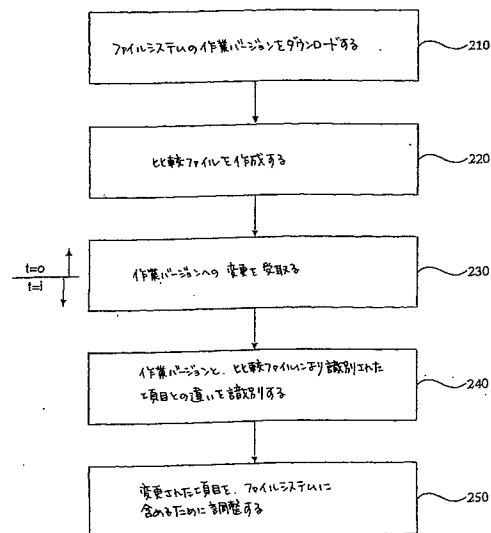


FIGURE 2

【 図 4 】

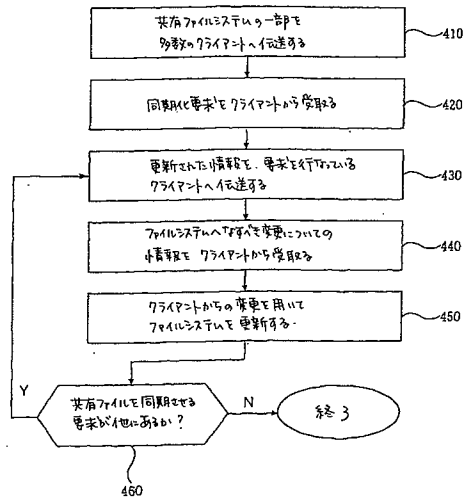


FIGURE 4

FIGURE 3

【图 6】

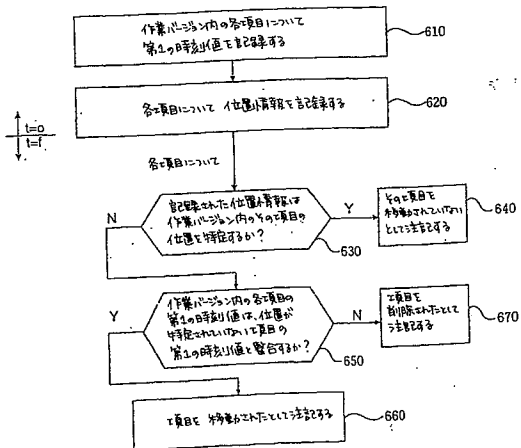


FIGURE 6

【図 7】

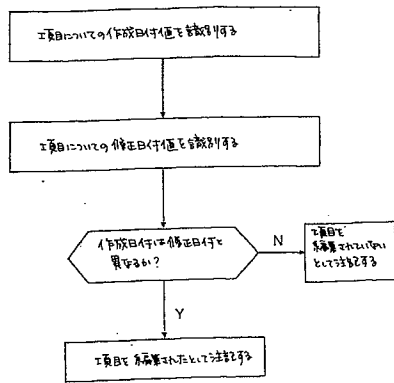


FIGURE 7

【図 8】

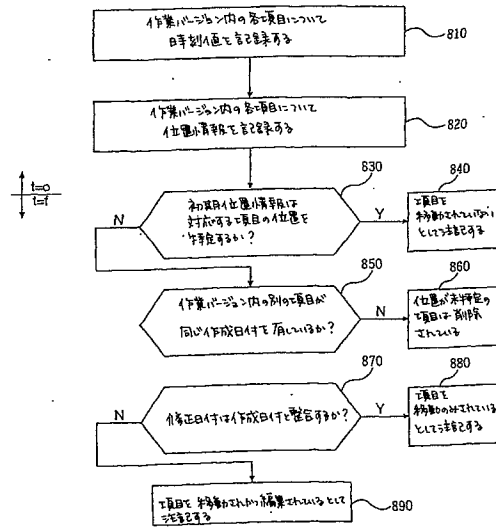


FIGURE 8

【図 9】

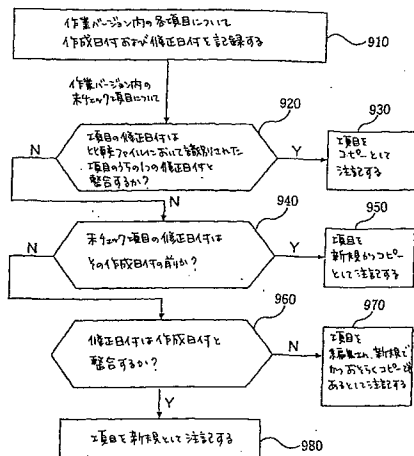


FIGURE 9

【図 10】

コンテナー A		コンテナー B	
ファイル名	必要取扱い	ファイル名	必要取扱い
A ₁	M ₁	B ₁	C ₁
A ₂	E	B ₂	E
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮
A _n	(N) ₁ C ₁	B _n	D

コーン インターフェイス

FIGURE 10

【図 11】

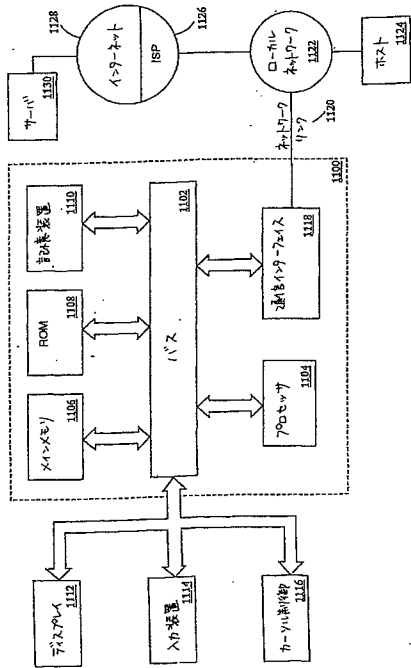


FIGURE 11

【国際公開パンフレット】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
5 December 2002 (05.12.2002)

PCT

(10) International Publication Number
WO 02/097623 A2

(51) International Patent Classification: G06F 9/46

(74) Agent: MAHAMED, Van et al.; HICKMAN
PALERMO TRIKORI & HICKMAN, LLP, 1600 Wil-
low Street, San Jose, CA 95125 (US).

(21) International Filing Date: PCT/US02/1681

(22) International Filing Date: 24 May 2002 (24.05.2002)

(25) Filing Language: English

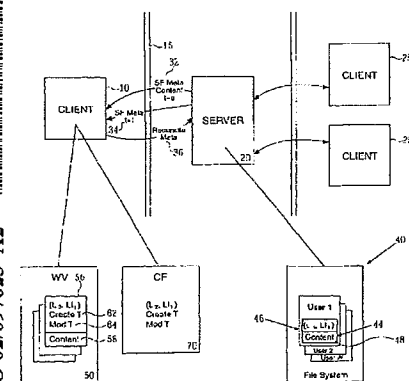
(26) Publication Language: English

(30) Priority Data: 09/866,143 25 May 2001 (25.05.2001) US

(71) Applicant: ORACLE INTERNATIONAL CORPORA-
TION (US/US); 501 Oracle Parkway, Redwood Shores,
CA 94065 (US).(72) Inventor: KIRNOS, Ilya; 1729 Woodland Avenue, Unit
M, Palo Alto, CA 94303 (US).(81) Designated States (national): AE, AG, AI, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU,
CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH,
GM, GR, GU, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC,
LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW,
MX, MY, NZ, OM, PA, PE, PG, PH, PT, RO, RU, SD, SE, SG,
SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN,
YU, ZA, ZM, ZW.(84) Designated States (regional): ARIPO patent (GH, GM,
KH, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),
European patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR,
GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent
(BF, BJ, CG, CI, CM, GA, GN, GQ, GW, ML, MR,
NE, NG, TD, TG).

[Continued on next page]

(54) Title: MANAGEMENT AND SYNCHRONIZATION APPLICATION FOR NETWORK FILE SYSTEM



(57) Abstract: An application is provided for managing files. Information about a file system is mapped to a comparison file. A working version is made from the file system. The application identify one or more operations were performed on the working version. The identified operations are recreated on the file system during a synchronization operation.

WO 02/097623 A2

WO 02/097623 A2 

Published:

— without international search report and to be republished
upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 02/097623

PCT/US02/16481

**MANAGEMENT AND SYNCHRONIZATION APPLICATION
FOR NETWORK FILE SYSTEM**

FIELD OF THE INVENTION

The present invention relates to an application for managing network files. In particular, embodiments of the invention pertain to detecting alterations to file systems for purpose of synchronization.

BACKGROUND OF THE INVENTION

When multiple file systems contain exactly the same content, the file systems are said to be "in sync". To keep file systems in sync, synchronization applications detect differences between file systems, and then perform operations to eliminate the differences. Typically, synchronization applications are used to synchronize the file systems of different computers that need to access different copies of the same files. A set of file systems that are to be synchronized with each other are referred to herein as a "synchronization set". Each file system that belongs to a given synchronization set is referred to as a "synchronized system".

The act of synchronizing a synchronization set is referred to as a synchronization operation. During each synchronization operation, a synchronization application typically attempts to detect when items have been deleted or added in any of the synchronized systems since the previous synchronization operation.

In general, synchronization applications seek to add to all synchronized systems those items that are detected as being added to any synchronized system since the previous synchronization operation. Similarly, synchronization applications seek to delete from all synchronized systems those items that are detected as being deleted from any synchronized system since the previous synchronization operation.

Typically, the synchronization application will not distinguish between (1) added items that are copies of other items and (2) added items that were created as originals. In addition, when a renamed item has been altered, it appears that the original item was deleted and a new item was added. Consequently, the original items may be deleted in all synchronized systems, and the new altered item will be added to all synchronized systems. Alternatively, the synchronization application may place both the altered item and the non-altered item together in the same file.

Under current synchronization techniques, if a user performs multiple operations on an item and then seeks to synchronize that item with another, the application will

WO 02/097623

PCT/US02/16481

detect that item as altered or new. The application will not be able to detect specific operations performed on the item. As a result, the synchronization operation may delete one file for another, or add the altered file to be stored with the original file in the same file system.

Synchronization operations often involve a significant amount of resource consumption. For example, when a synchronization application detects the addition of a new file to one synchronized system, data transfer of the contents of the new file to all other synchronized systems is required. If the file is large and/or the number of synchronized systems is large, the resource consumption may be significant.

Another problem with current synchronization techniques is that new or replaced files do not retain metadata information from prior to their transfer or recreation on the file system. Thus, if a file created at time T1 is altered, the fact that the file was originally created at time T1 will be lost by when the synchronization application treats the altered file as a new file, and the original file as a deleted file.

SUMMARY OF THE INVENTION

Embodiments of the invention provide an application that can detect one or more operations performed on a first file system that is to be synchronized. The synchronization application updates a second file system using the detected operations of the first file system.

An embodiment of the invention is capable of detecting operations in the first file system, including copying an item, moving an item, creating a new item, deleting an item, and editing an item. An embodiment of the invention also detects multiple operations performed on the first file system. The detected operations may be recreated on the second file system during a synchronization operation.

Synchronization techniques described with embodiments of the invention incur less overhead than other synchronization processes in use. Furthermore, embodiments of the invention provide for synchronization techniques that preserve metadata information about synchronized files, in contrast to other synchronization processes that incur loss of such information.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements.

WO 02/097623

PCT/US02/16481

FIG. 1 is an overview of a system architecture, under an embodiment of the invention.

FIG. 2 is a flowchart describing synchronization on a terminal, under an embodiment of the invention.

FIG. 3 is a flowchart describing synchronization of a file shared by many users on a terminal of the system, under an embodiment of the invention.

FIG. 4 is a flowchart detailing synchronization of a shared file system on multiple terminals, under an embodiment of the invention.

FIG. 5 is a flowchart detailing detection of multiple operations and compound operations on a working version of a file system, under an embodiment of the invention.

FIG. 6 is a flowchart for identifying moved or deleted items during a synchronization operation.

FIG. 7 is a flowchart for identifying edited items during a synchronization operation, under an embodiment of the invention.

FIG. 8 is a flowchart for identifying one or more operations for an item that was edited and/or moved or deleted, under an embodiment of the invention.

FIG. 9 is a flowchart for identifying items that have been created as new or copied from other items, and possibly edited, under an embodiment of the invention.

FIG. 10 illustrates a user-interface for use with an embodiment of the invention.

FIG. 11 is a hardware block diagram for use with an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

A method and apparatus for managing files is described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

Among advantages provided, embodiments of the invention enable a synchronization operation to be performed that identifies specific acts performed to file items that belong to synchronized systems. The specific operations are recreated on corresponding file items of other synchronized systems during the synchronization operation. As a result, complete transfers of file items may be avoided in many instances where such file items were merely altered or moved in some way. The result is that communication resources are conserved. In addition, corresponding file items of the other

WO 02/097623

PCT/US02/16481

synchronized systems may be updated without losing metadata information for those items.

A. Functional Overview

In an embodiment, one or more client terminals can access a file system on a server. One or more files can be downloaded from the file system and manipulated on the client terminal. In particular, a user may perform certain acts on the content of the downloaded file, including editing documents, deleting items, creating new documents for the file, moving items or copying items within the file, or combinations of these acts. Under an embodiment, a management system detects the acts performed on the downloaded file. The management system then synchronizes the downloaded file with a corresponding portion of the file system.

As described herein, the file system is part of an overall management system that keeps numerous files for multiple clients. A client may download only a portion of the file system. The portion of the file system may include items such as directories, sub-files, applications, executables, documents, and individual resources of different data types.

When portions of the file system are downloaded by a client, the resulting local file is referred to as a working version. The working version copies the items from the portion of the file system selected to be downloaded for the client. Information is recorded in a comparison file about the working version when it is created. The comparison file may also include information about the portion of the file system downloaded. This information includes metadata information that can subsequently be used to identify file items, as well as modifications made to the working version after it is created. After the working version is modified, the working version can be synchronized with the portion of the file system used to download the working version. Information recorded in the comparison file is used to detect the changes made to the working version.

As used herein, reference to the term "item" means data structures that can be maintained and/or managed within file systems. As noted, items include directories, files, applications, executables, documents, and individual resources of different data types. The item may include a document or resource of a particular data type. For example, a first item may be a word processing document, and a second item may be a folder that stores the document with other resources.

WO 02/097623

PCT/US02/16481

Under an embodiment, information included in the comparison file is primarily metadata information. The metadata information may include location information for a particular item, creation times, modification times, item size, and file names.

A location is identifiable by a memory address and computer location. Location information refers to data that can be used to identify the memory location of the item on a computer. Location information may include a file or resource name. Location information may also include a file path locating a particular item in memory.

Embodiments of the invention include a system and method of managing files. According to one embodiment, after a first synchronization operation, information is mapped from a file system to a comparison file. The information includes information about the status, after the first synchronization operation, of a first item. For example, the information in the comparison file may indicate that the first item was located at a first location after the first synchronization operation. In addition to mapping the file system information to the comparison file, a working version of the file system is made. Initially, the working version indicates that the first item is at the first location. The information contained in the working version is modified to reflect any changes made to the status of the first file after the first synchronization operation. During a second, subsequent synchronization operation, the location indicated in the working version of the file system is compared with the location indicated in the comparison file to determine whether, during the interval between the first synchronization operation and the second synchronization operation, the first item has been moved.

In an embodiment, the first item in the file system may be moved to a new location that is identifiable by new location information. Therefore, the file system does not have to recreate the first working item if the first working item is moved. Rather, the file system can move the first item corresponding to the first working file to a corresponding location in the file system. In contrast to previous synchronization applications, such an embodiment of the invention does not require the first working item to be recreated as a new part of the file system just because it was moved on the working version. As a result, communication resources are preserved because a data transfer of the contents of first working item is not necessary. Furthermore, metadata information for the first item in the file system is preserved.

Another embodiment of the invention maps information about a file system to a comparison file. A working version is made of a portion of the file system. During a synchronization operation, the comparison file and the working version are used to determine whether items have been copied since the last synchronization operation.

WO 02/097623

PCT/US02/16481

Another embodiment of the invention provides a computer system that can operate a network management application. The computer system includes a network interface to exchange communications with a second computer. The communications are to create a working version of a file system portion accessible on the second computer. The first computer includes a memory that stores the working version. A processor on the first computer records a creation time for at least one working item in the working version, where the first working item originates from a first item of the file system. The processor subsequently uses the creation time to determine if an operation was performed on the first working item.

Among other advantages, embodiments of the invention can detect if an item was moved to a new location after the working version was created, if an item was copied from another item existing in the working version when it was created from another file, or if an item was copied from an item added to the working version subsequent to its creation. Other operations that can be detected under an embodiment of the invention include if an item was edited, or deleted from the working version. An embodiment of the invention can also detect multiple operations performed on or for an item in the working version.

The result is that a portion of a file system used to create the working version can be updated to reflect subsequent changes in the working version. However, items in the file system that are subsequently updated by working version items do not need to be entirely replaced, or be recreated by the working version items. Rather, a synchronization method or application can update the file system to reflect changes to corresponding items of the working version, or to additions of items to the working version. Another advantage is that file system items that are operated on in the working version can maintain information that tracks their origin. As a result, the file system can be updated to reflect only those operations performed on the working version.

B. System Overview

FIG. 1 illustrates a system for managing files shared between computers, under an embodiment of the invention. The system includes a first terminal 10 coupled to a server 20 via a network 15. A plurality of other terminals 25 may also be coupled to server 20 over network 15. The first terminal 10 may be operated as a client that communicates with server 20. In an embodiment, a client application is operable on first terminal 10 to manage file items and resources that are shared with server 20.

WO 02/097623

PCT/US02/16481

A user may operate first terminal 10 to access a file system 40 containing one or more resources and other items from server 20. The original version of the file system 40 may remain on server 20 while the user works on a borrowed or working version of a portion of the file system 40. A system such as described by FIG. 1 enables the user to work locally on items accessed from a remote server 20, then update file system 40 on the remote server to reflect changes made on first terminal 10 to those items.

In an embodiment, the user on terminal 10 can perform operations on items accessed from server 20. These operations may include editing content, deleting particular items retrieved from the server, moving items to new locations, copying items retrieved from server 20, and adding new items for subsequent inclusion on server 20. In addition, an embodiment of the invention allows a user to update server 20 to reflect combinations of operations performed on items. An embodiment of the invention reduces possible combinations of operations performed by the user into equivalent compound operations. The equivalent compound operations may include editing and copying an item, creating new items then editing them and/or copying them, as well as editing existing items and then editing them.

In an embodiment, terminal 10 exchanges communications with server 20 using a network interface 12. In one implementation, network interface 12 enables Internet Protocol (IP) communications, and specifically Transport Control Protocol (TCP/IP) for enabling communications over networks such as the Internet. Alternatively, embodiments of the invention may signal communications between computers over networks such as Local Area Networks (LANs) and other types of Wide Area Networks (WANs).

The server 20 may be used to store or otherwise manage a file system 40. In an embodiment, file system 40 includes a plurality of portions, where each portion is associated with a user or an account. A first portion 46 of file system 40 may be a file stored on server 20, that is accessible to first terminal 10, or to a user of first terminal 10. The first portion 46 may include a plurality of items, such as files and resources of particular data types.

A first item 44 of first portion 46 is identified in FIG. 1. For illustrative purposes, first item 44 is assumed to be a resource such as a document. Alternatively, first item 44 could be a file containing other items. The first item 44 includes or is otherwise associated with metadata information and content. The metadata information of first item 44 may identify a particular location (L1) on a memory (not shown) of server 20. The metadata information of first item 44 may also include a location identification information (LI1) used to locate the first location (L1) on server 20. Since first item 44 is

WO 02/097623

PCT/US02/16481

assumed to be a resource, first item 44 also includes a content associated with the metadata information.

In an embodiment shown, first terminal 10 receives a first communication 32 from server 20, signaled over network interface 12 and network 15. The first communication 32 includes first portion 46 of file system 40. In one implementation, a user operating first terminal 10 has access rights to first portion 46. The access rights enable the user to download or otherwise retrieve some or all of first portion 46, including first item 44. The user can make a working version 50 of first portion 46 after receiving first communication 32. The working version 50 includes content from items in first portion 46. Certain metadata information for working version 50 may be transferred from server 20 and included in first communication 32. Other metadata information may be generated on first terminal 10 when working version 50 is made. Metadata information transferred over from the file system may include, for example, location information such as file paths and names to locate certain items.

Data signaled with first communication 32 may be used to generate working version 50, including at least a first working item 56. The first working item 56 originates from first item 44 of file system 40. In an embodiment, the first working item 56 originates from first item 44 because a content portion 58 of first working item 56 is copied from a corresponding content portion 48 of first item 44.

Metadata information that may be carried over from file system 40 includes the first location information (LI1) of first item 46. The first location information (LI1) may be used to identify a second location (L2) for first working item 56 on first terminal 10. For example, the first location information (LI1) may include a file path and a name. The file path may be recreated on working version 50 to enable first working item 56 to be located at the second location (L2). The name may be transferred over also as additional location information. In many applications, the name is an inclusive portion of the file path.

When working version 50 is generated on first computer 10, new metadata information is recorded. The new metadata information may include time values that mark certain events for first working item 56. In an embodiment, a first time value 62 may correspond to a creation time for the first working item. A second time value 64 may correspond to a modification time for first working item 56. The first time value 62 and second time value 64 are initialized upon or just after working version 50 is generated on first computer 10. As an example, a user may download a word processing document as first working item 56. When the document is downloaded, first time value 62 (creation

WO 02/097623

PCT/US02/16481

time) and second time value 64 (modification time) are recorded by an operating system (or other application) on first terminal 10. For example, first terminal 10 may run a WINDOWS type operating system that automatically records creation time values and modification time values when first working item 56 is generated. The creation time is a value assigned to a particular item marking the time of its creation on a particular computer system. The creation time is stored as a static value that can be used subsequently identify a corresponding working item, even if that working item has a new address or a new name. The modification time is a value associated with the working items to mark the last instant that the item was edited or created. The modification time can therefore change after working version 50 is downloaded from file system 40.

In an embodiment, first computer 10 maintains or otherwise accesses a comparison file 70 to store metadata information. The metadata information stored in comparison file 70 may include new metadata information recorded when first working item 56 is generated on first computer 10, as well as certain metadata information that may be carried over from first item 44 of file system 40.

In the example provided, comparison file 70 stores the first location information (L11) of the first working item 56 and the first item 44, the second location (L2) of the first working item 56, the first time value 62 (creation time) of the first working item 56, and the second time value 64 (modification time) of the first working item 56. The first location information (L11) is transferred from server 20, while other metadata information in comparison file 70 is created with generation of working version 50. The metadata information at the initial moment for items in working version 50 are generated and stored in comparison file 70. As shall be described in greater detail hereafter, this metadata information is used to identify the specific operations that are performed on first working item 56 after the working version 50 is made. By knowing the specific operations, synchronization may be performed more efficiently.

The operations that comparison file 70 may be used to detect include the operations of editing items, moving items, making new items, copying items, deleting items and combinations thereof. Comparison file 70 provides access to metadata information for each item signaled with first communication 32. Subsequent to operations being performed on working version 50, an embodiment of the invention provides that metadata in items of working version 50 are compared against comparison file 70. The comparison of metadata information is used to detect the operation(s) performed on working version 50, for purpose of determining differences between working version items and file system items. In making the comparison, items in working version 50 may

WO 02/097623

PCT/US02/16481

be detected as having metadata information that are different from metadata of corresponding items recorded by comparison file 70. In addition, items in working version 50 may be detected as not having a corresponding item identified by comparison file 70. The differences identified when making the comparisons are noted and used to synchronize the working version 50 with first portion 46 on file system 40.

Under an embodiment of the invention, first portion 46 is a shared file accessible to other terminals 25 from server 20. It is possible that first portion 46 is changed by another computer after portions of it are signaled to the first terminal 10. The other terminal 25 may, for example, access and operate on items in first portion 46 so that first portion 46 is changed from the time it is signaled to first terminal 10. In order to make the comparison for identifying the changes in working version 50 with shared file system 40, a second communication 34 is signaled to first terminal 10 from server 20. The second communication 34 includes metadata information as existing on server 20 at the moment synchronization is to be performed with first terminal 10. In an embodiment, second communication 34 is signaled to first terminal 10 upon a synchronization request being made from the first terminal 10.

In an embodiment, first terminal 10 performs the synchronization operation. The synchronization operation may compare metadata information between changed or added items of working file 50 and items of first portion 46. The changed or added working version items are the result of the first terminal's users performing one or more operations on working version 50. Changed or added file system items are the result of other users performing one or more operations on their versions of file system 40. The differences between working version 50 and items of first portion 46 are identified and reconciled by the user of first terminal 10. The differences are recorded as reconciled metadata information. In an embodiment, a third communication 36 is used to signal the reconciled metadata information from first terminal 10 to server 20. The reconciled metadata information may be signaled to server 20 to cause server 20 to perform one or more operations that update file system 40 so as to reflect changes from operations performed on working version 50. Furthermore, the reconcile information may be displayed by the user 50 so that the user can select changed or altered items that will be used to update file system 40.

In another embodiment, first portion 46 is not shared with other users, but only with the user of first terminal 10. As such, second communication 34 may not be necessary. Rather, comparison file 70 is used to perform the synchronization operation and to identify reconciled metadata information. The reconciled metadata information is

WO 02/097623

PCT/US02/16481

then signaled to server 20 after the synchronization operation is performed on first terminal 10. The reconciled metadata information is signaled to server 20 to cause it to update file system 40 with changes of working version 50.

WO 02/097623

PCT/US02/16481

C. Synchronization Operations

FIG. 2 illustrates a method for making working version 50, and subsequently synchronizing working version 50 (FIG. 1) with a corresponding portion of file system 40. Reference to components of FIG. 1 is intended to communicate exemplary components for use with that embodiment. In an embodiment such as described with FIG. 2, it is assumed that first file system 40 is not shared with other users.

In a step 210, a working version of file system 40 portion is downloaded onto first terminal 10. For example, first terminal 10 may connect with server 20 over the Internet. The user of first terminal 10 may have an account to identify first portion 46 of file system 40. The first portion 46 of file system 40 can be selected by the user to be downloaded onto first terminal 10.

In step 220, comparison file 70 is generated when working version 50 is made. The comparison file records initial metadata information of working version 50. Some of the metadata information may also be transferred from items of first portion 46 of file system 40. Steps 210 and 220 are performed at $t=0$, prior to any operations that may affect working version 50. Steps 230-250 occur after some time $t=i$, so that the user may have performed an operation on working version 50. At that time, the user is making a request to synchronize working version 50 with file system 40.

In step 230, differences are identified between modified working version 50 and working version 50 at the time comparison file 70 was created. The differences may be referred to as delta items. The delta items include items in working version 50 at the later time that are new, copied, moved, or modified. The delta items may also include items identified by comparison file 70 that have no correspondence or counterpart in working version 50. For example, first working item 56 may be operated on through edits and moves, in which case it is a delta item in working version 50. Alternatively, comparison file 70 may identify working item 56, but working item 56 may have been deleted from working version 50. In this case, first working item 56 is a delta item in comparison file 70. Similarly, other working items may be copied or added to working version 50 after comparison file 70 is made, in which case those items are identified as delta items in working version 50.

In step 240, differences between items in working version 50 and comparison file 70 are identified. As discussed, these differences are also referred to as delta items.

In step 250, differences identified between working version 50 and items identified by comparison file 70 are reconciled. To reconcile, delta items may be selected for instructing file system 40 to be updated. For example, if a delta item is an edited

WO 02/097623

PCT/US02/16481

version of first working item 56, then the selection specifies whether file system 40 is to include the edited or the original version of first item 44. If a delta item is an added item (such as a new or copied item) to working version 50, then the selection determines whether file system 40 is to keep those additions. If the delta item is first working item 56 moved to a new location, then the selection determines whether file system 40 is to use new location information for first item 44, or whether the file system is to maintain the old location. If the delta first working item 56 deleted from working version 50, the selection specifies whether file system 40 is to delete first item 44. Similar methods may be performed with combinations of operations, as detailed elsewhere in this application.

FIG. 3 details a method for synchronizing working version 50 with file system 40 on server 20, when system 40 is shared with other computers, under an embodiment of the invention. In FIG. 3, working version 50 is made as described with an embodiment of FIG. 2. The working version 50 is downloaded from file system 40 in step 310. Comparison file 70 is created to record metadata information about working version 50 and file system 40 in step 320. Changes are made to working version 50 in step 330. Differences between the working version 50 and items identified by comparison file 70 are identified in step 340. These items, referred to as delta items, may include working version items that have at some point been moved, deleted, edited, are added through one or more operations.

In step 350, the user of first terminal 10 makes a synchronization request with server 20. By this time, working version 50 may have been modified from its original state by one or more operations.

In step 360, new information is received about file system 40 on first terminal 10. The file system 40 may have been accessed and altered by other terminals since the time working version 50 was generated on first terminal 10. Therefore, new information about file system 40 may identify changes made to items of the file system 40 by other users. In one implementation, information about the file system 40 is in the form of metadata, and may be particular to items of first portion 46 downloaded by the user of first terminal 10. The metadata information may include location information of file system items that correspond to the downloaded items. In addition, the new metadata information about file system items may include time values. For example, create time and modification time values of the file system items at the time the synchronization is requested may be signaled to first terminal 10 for purpose of determining delta items of file system 40.

In step 370, differences, or delta items, are detected between the update file system 40 and the file system at the time the working version was made. These delta

WO 02/097623

PCT/US02/16481

items are identified by new metadata information received in step 360 compared to items identified by comparison file 70 when the comparison file was created in step 320. The delta items identified in this step may be identified by either comparison file 70 or by the new metadata information received about file system 40. Delta items identified by new metadata information about file system 40 may correspond to items that were moved or edited by other users. In addition, delta items of file system 40 may include items added to first portion 46 by other users, either as new items or copies of other items. Delta items identified by comparison file 70 include items deleted from file system 40 after working version 50 is made on first computer 10.

In step 380, selections are made for delta items identified in step 340 and in step 370. The selections may be made by a user. The selections may specify delta items of comparison file 70, working version 50, and file system 40. For each delta item, the selection may determine whether to keep that delta item or not.

In step 390, conflicts between differences identified in steps 340 and 370 are detected and resolved. For example, an item in the working version 50 may be edited, so that it is identified as a delta item when compared to a corresponding file system item at the time working version 50 was made. That file system item identified in comparison file 70 may be subsequently changed by another computer having access to server 20. Thus, two delta items may be associated with the same item identified by comparison file 70. In an embodiment, the user of first terminal 10 can choose which of the two delta items should be used for inclusion in file system 40.

Alternatively, conflict selections between delta items may be made through a conflict protocol that selects whether each delta item is to be incorporated into the synchronized file system 40.

In step 395, the selected delta items are used to update file system 40. Each delta item identified in step 340 and 370 may be omitted or included in the file system's update. The user can select between delta items in conflict.

FIG. 4 illustrates a method for operating server 20 under another embodiment of the invention. In an embodiment such as described with FIG. 4, multiple users are assumed to access shared file systems on server 20. Reference is made to FIG. 1 for descriptive purposes. The first terminal 10 is assumed to be making the synchronization request. Portions of file system 40 is shared with other clients 25 who can access server 20.

WO 02/097623

PCT/US02/16481

In step 410, a portion of the shared file system 40 is signaled to first terminal 10 and clients 25. Each client may be operated separately to access and receive portions of the shared file system.

In step 420, a synchronization request is received from first terminal 10. The synchronization request may correspond to a user wishing to implement their changes to the file system 40. The users may also wish to receive any changes entered from other users who downloaded the portion of the file system 40.

In step 430, updated information about the file system 40 may be signaled to the client requesting the synchronization. The file system 40 may be updated from the time that client downloaded the file system to include changes entered from clients 25.

In step 440, server 20 receives information about changes that are to be made to file system 40 as a result of operations performed on working version 50. The changes may be the result of operations such as edits, additions (new items and copies), deletions, and moves.

In step 450, file system 40 is updated using changes signaled from first terminal 10 (the client making the synchronization request). The updated changes may be selections decided by a particular user after performing one or more operations on that terminals working version of the file system 40.

In step 460, a determination is made as to whether any other requests are made or will be made from other terminals that access file system 40. If there are other requests to synchronize, then step 430-460 are repeated for the next client making the request. Under such an implementation, each client having access to the shared file system 40 makes additional changes to it. The changes and alterations made by other users are incorporated in file system 40 when the synchronization request is made. Therefore, file system 40 changes after each synchronization operation with one of the clients, so that the next client is synchronizing with a previously updated file system 40.

D. Detecting Operations on Working Files

With reference to exemplary components of FIG. 1, embodiments of the invention enable synchronization between items of the working version 50 and items of the file system 40, even after the items undergo multiple and different types of operations. The operations that can be performed on working version 50 may be characterized as a primary operation or a compound operation. Under an embodiment, multiple operations performed on an item can be detected as one of a set of equivalent compound operations.

WO 02/097623

PCT/US02/16481

In an embodiment, the primary functions are edit, delete, copy, move and create new. The edit operation results in a content of an item in working version 50 being changed. The delete operation causes an item to be removed from the working version 50. The copy operation recreates the contents of an item in working version 50 as a new or added item. The move operation causes an item located in one location of the working version 50 to be given a new location. A location may be defined by a name, a memory address and a memory unit. Thus, the move operation may be performed to move an item to a new folder location, to rename an item, or to move the item to a new memory unit. The create new operation is performed on the working version 50 to create an additional item to the working version 50.

A compound operation is a combination of multiple operations performed on the working version 50, to create and/or affect a working version item. In contrast to embodiments of the invention, previous synchronization systems are able to detect performance of some of the primary operations, but are unable to detect certain primary operations, or combinations of operations. Advantages provided by embodiments of the invention enable detection and synchronization of all the primary operations, as well as combinations of multiple operations performed on individual items of working version 50.

Analytical expressions for describing operations of file management may be described using the format aOb, where the capitalized letter stands for the operation, an item preceding the operation represents the source for the operation, and an item following the operation represents the destination for the operation. To summarize the primary operations:

Ex -- Edit file X
 Dx -- Delete file X
 Nx -- Create new item X
 xMy--X is moved to Y
 xCy--X is copied as Y

Under an embodiment of the invention, compound operations can be reduced and abstracted to a finite number of equivalent compound operations. Some examples of principles used in making these abstractions include: (1) if an item is deleted, previous operations performed on that item can be ignored; (2) multiple moves of an item can be treated as one move, from the initial source to the final destination; and (3) any move operation performed on a combination of operations can be analyzed in any order with respect to other operations, so assuming a move is performed before another operation

WO 02/097623

PCT/US02/16481

provides a true and simplified result. Using these principles, it can be assumed that any working item undergoes one of nine possible operations or combinations of operations, where the combinations of operations are equivalents of other operation combinations. The operations performed on items of working version 50 can be replicated for file system 40 as either one of the five primary operations, or one of four equivalent compound operations. In an embodiment, the four equivalent combinations of operations are:

ExMy--Edit X and move it to Y

(Nx)Cy--Create X and copy it as Y

E((Nx)Cy)--Create X, copy it as Y, and edit Y

E(xCy)--Copy X as Y, and edit Y

Parentheticals are to be performed first in any equivalent compound operation.

FIG. 5 illustrates a method for detecting operations performed for items in working version 50 at the time a user requests to synchronize the working version 50 with file system 40, under an embodiment of the invention. In an embodiment such as shown, there are ten possible outcomes for each item in the working version 50 at the time of synchronization: unchanged, five prime operations, and four equivalent combinations of operations.

In step 502, working version 50 is created from a portion of file system 40. In step 504, a comparison file 70 is made that includes information about the working version 50. Both steps 502 and 504 are assumed to take place before any operations are performed for the working version items (i.e. at $t = 0$). At a subsequent moment, the user requests synchronization with the file system (i.e. at $t = f$). From between $t=0$ and $t=f$, the user may perform one or more operations that alters working version 50.

Upon receiving a synchronization request, a determination is made in step 506 as to whether an item identified and located by comparison file 70 has a same location as a corresponding item in working version 50. Initially when comparison file 70 is made, the location of each item is recorded. Thus, step 506 determines whether an item identified in comparison file 70 can still be located using location information initially recorded for that item.

If step 506 determines that the item identified by comparison file 70 still has the same location within working version 50, step 508 follows with another determination as to whether that item was edited subsequent to being recorded in comparison file 70. If step 508 determines the item was not edited, then step 510 concludes that the particular

WO 02/097623

PCT/US02/16481

item was unchanged in working version 50. If step 508 determines that the particular item was edited, then step 512 notes the item identified by comparison file 70 as edited.

If step 506 determines that the item identified by comparison file 70 was not located by information recorded for that item, then step 514 makes a determination as to whether the item was moved. If the determination is that the item was not moved, step 516 notes the item as being deleted. If the determination is that the item was moved, then step 518 notes the new location of the item in working version 50. Then a determination is made in step 520 as to whether the moved item was also edited. If the determination is positive, then the item is marked as moved and edited in step 522.

Step 524 makes a determination as to whether any more items identified by comparison file 70 remain to be checked. Step 524 follows step 510 if the previous item was determined to be unchanged. Step 524 follows step 512 if the previous item was noted as being edited. Step 524 follows step 516 if the previous item was noted as deleted. Step 524 follows step 520 if the previous item was noted as moved. Step 524 follows step 522 if the previous item was noted as moved and edited. If step 524 determines items remain that are identified by comparison file 70, and that these items have not been checked, then step 526 provides the next item identified by comparison file 70 is to be checked. For the next item, the method is repeated beginning with step 506.

If step 524 determines no items identified by comparison file 70 remain to be checked, step 528 determines if any items remain in working version 50 unchecked. The unchecked items are working version items that were not checked as a result of comparison file items being checked in steps 506-524. If there are no unchecked items in working version 50, the method is done. Else, the remaining items in working version 50 are marked checked in step 530.

Step 532 determines whether an unchecked item in working version 50 is a copy. If the determination is positive, step 534 notes the item as a copy. Step 536 determines whether the copied item was copied from another item that was created new in working version 50.

If the determination in step 536 is negative, step 538 determines whether the copied item was also edited after being made in step 538. If the determination in step 538 is positive, step 540 notes the copied and edited.

If the determination in step 536 is positive, the item is marked as new and copied in step 542. In other words, the item is noted as being copied from another item that was created as an original, after working version 50 was created from file system 40. In step

WO 02/097623

PCT/US02/16481

544, a determination is made as to whether the new and copied item was also edited. If the determination in step 544 is positive, the item is noted as new, copied and edited.

If in step 532 the determination is that the item is not a copy, then in step 548 the item is noted as being new.

Step 554 determines whether any items marked as unchecked remain in working version 50. Step 544 follows one of these steps: if the determination in step 538 is negative, so that the item is noted only as being copied; step 540 if the item is determined to be copied and edited; step 548 if the item is determined to be only new; if the determination in step 544 is negative, so that the item is determined to be new and copied; and step 546 if the item is determined to be new, edited and copied. If step 554 determines that items are remaining that are unchecked in working version 50, step 556 iterates to the next unchecked item. Then the method is repeated for the next item, beginning with step 532. If step 554 determines that unchecked items are not remaining in working version 50, the method is done.

As shown by an embodiment of FIG. 5, a synchronization operation may detect ten possible outcomes for each item being synchronized. Each item may be determined as being unchanged since being downloaded (step 510). Otherwise, each item being synchronized may be determined to be a result of one or more operations performed by a user after working version 50 is created. Five prime operations are detected: edit (step 512), move (step 518), delete (step 516), create new (step 548), and create copy (step 534). In addition, four compound operations are detected: move and edit (step 522); new and copy (step 542); new, edit and copy (step 546), and copy and edit (step 540).

In an embodiment, a special case may arise where a working version item is deleted, and then recreated with the same name and location information. Such an item could be identified as a new item rather than a moved item if step 506 incorporates a check for the special case. Specifically, an identification such as the recreated item's creation time value may be used to check that the item was not subject to deletion and recreation in step 506.

FIGS. 6-9 illustrate flow charts providing additional details for detecting operations performed in FIG. 5, under embodiments of the invention. FIG. 6 is a method for determining whether an item is moved or deleted. FIG. 6 may correspond to steps 514-518 of FIG. 5, under an embodiment of the invention.

In step 610, a first time value is recorded for each item in working version 50 when working version 50 is made on first terminal 10. In an embodiment, the first time value may correspond to a creation time of an item. The creation time is a property

WO 02/097623

PCT/US02/16481

attached to items under certain operations systems, such as WINDOWS. The creation time may record a time value for an item created on first terminal 10 when that item is downloaded from another computer. Thus, when an item of working version 50 is downloaded from file system 40, the first terminal 10 may record the creation time of that item. The creation time may be significant to a thousandth of a second, or to several magnitudes greater.

In step 620, location information is created for each item when working version 50 is made. The location information may correspond to segments of file paths or names that can be used to locate an item either on file system 40 or on working version 50. Both steps 610 and 620 occur at $t = 0$, corresponding to when working version 50 is made, and before any operations are performed. The first time value and the initial location information may be located in comparison file 70.

After step 620, the flowchart forwards to when a synchronization request is made, or at $t = f$. Step 630 determines whether location information recorded initially (at $t = 0$) locates the item in working version 50 at $t = f$. If the determination in step 630 is positive, then step 640 records the item as not being moved. If the determination is negative, step 650 follows. Step 650 determines whether any item in working version 50 at $t = f$ has a corresponding first time value that matches the time value recorded for the unlocated item in step 610. In one embodiment, other items in working version 50 may be checked for a creation time that matches the creation time of the unlocated item.

Given that the creation time may be carried out past a thousandth or even a millionth of a second, another item in working version 50 having the same creation time as the missing item can be assumed to be the unlocated item in a new location. If the determination in step 650 is positive, the item having the same time value is recorded as being moved in step 660. If step 650 determines that no item in working version 50 has the creation time of the missing item, step 670 notes that item as being deleted.

FIG. 7 is a flowchart for determining whether any item in working version 50 has been subject to an edit operation, under an embodiment of the invention. A method illustrated by FIG. 7 may correspond to steps 508, 510 and 512 of FIG. 5. In step 710, a first time value is identified in working version 50. The time value may correspond to that item's creation time. In step 720, a second time value is identified for the same item, corresponding to that item's modification time. As mentioned, both the creation time and modification time are time values that are automatically recorded by operating systems such as WINDOWS. Both of these time values may be significant to a thousandth of a second, or even to much greater orders of accuracy. Thus, for an embodiment such as

WO 02/097623

PCT/US02/16481

shown by FIG. 7, both the creation time and modification times are assumed to be unique to that item.

In step 730, a determination is made as to whether the modification time is different than the creation time. When an item is created, either as an original, copy or a download, an embodiment provides that the creation time and modification time are the same. Thus, if the creation time and modification times are different, step 740 notes the item as edited. Else, step 750 notes the item as not edited.

It is possible the creation time and the modification time are not initially exactly the same, but within a range of one another. An embodiment may check to determine if the modification time is outside the range of the creation time.

FIG. 8 is a method for identifying the compound operation of edit and move, under an embodiment of the invention. In one implementation, a method as shown by FIG. 8 may be used as sub-steps for steps 514, 518, 520 and 522 of FIG. 5.

In step 810, multiple time values are recorded for each item in working version 50 when the working version is downloaded from the file system 40. As noted in other embodiments, a first one of the recorded time values corresponds to a creation time. The creation time may automatically be created by the operating system of the computer using the working version 50. The creation time and the modification time may each be recorded in comparison file 70, and associated to a corresponding item.

In step 820, location information is recorded for each item when the working version 50 is made. The location information may include segments of file paths that can locate the item in working version 50. The location information may also include a name of the item. The initial location information for each item in the working version may be recorded in comparison file 70.

In step 830, a determination is made as to whether initially recorded location information can subsequently locate a corresponding item working version 50. If the location information does locate the corresponding item, the item is recorded as not moved in step 840. If the location information recorded initially does not locate the corresponding item, then in step 850 another determination is made. Step 850 determines whether another item in the working version 50 has the same creation time as the unlocated item. If this determination is negative, then step 860 notes the unlocated item as deleted.

Otherwise, step 870 makes a determination as to whether the modification time matches the creation time for that item. If the determination in step 870 is positive, the

WO 02/097623

PCT/US02/16481

item is noted only as being moved in step 880. If the determination in step 870 is negative, the item is noted as being moved and edited in step 890.

FIG. 9 illustrates a process for detecting one or more operations on the unchecked items in working version 50, under an embodiment of the invention. With a process shown by FIG. 9, the operations performed on the unchecked items may include at least two operations from a group consisting of create new, copy and edit. In an embodiment, a process such as shown by FIG. 9 may form sub-steps of steps 532-546 in FIG. 5.

The steps 910-980 are performed on individual unchecked items in working version 50, at the time synchronization is requested. Steps 910-980 assume certain other steps have already been performed to detect other operations that may have been performed on working version 50. Specifically, 910-980 are performed on unchecked items in working version 50. As illustrated with FIG. 5, the unchecked items are items left over after items identified by comparison file 70 are compared to items of working version 50. The unchecked items can therefore be assumed to have been created after the working version 50 was made. Thus, unchecked items are a copy and/or a new item. The unchecked items may also have been edited after being created.

In step 910, time values are recorded for each unchecked item in the working version. For items created subsequent to working version 50 being downloaded, the creation time may correspond to when a user created that item and stored it with downloaded items in working version 50. The creation time should provide each unchecked item in working version 50 with a unique identifier. In addition, the modification time for each item in working version 50 is recorded. The modification time changes each time the corresponding item is edited. However, if the item is not edited, the modification time should be the same or very close to the creation time for that same item. In an embodiment, it can be assumed that the creation time for each item in working version 50 matches the creation time stored for that item in comparison file 70.

Step 920 determines whether the modification time for each unchecked item matches the modification time of one of the items stored in comparison file 70. An embodiment provides that modification time of a copy is the same as the modification time of its original. This feature may be implemented through an application operated on first terminal 10. In an embodiment, first terminal 10 operates an operating system that includes this attribute or feature. An example of such an operating system is WINDOWS type operating system.

If the determination is positive, step 930 provides that the item is noted as being created as a copy of another item originally downloaded from file system 40. The item

WO 02/097623

PCT/US02/16481

can be assumed to not have subsequently been edited because the edit operation alters the modification time. If the determination is negative, then step 940 follows.

Step 940 determines whether the modification time of the unchecked item is before the creation time. If the modification time is after the creation time, step 950 notes the item as a copy of another item. This is because a copy of another item keeps the original's modification time, but is assigned a new creation time when created. Step 940 cannot be used to detect whether an item created as a copy was subsequently edited, as that would alter the modification time to be after the creation time.

If the modification time is after the creation time, step 960 makes a determination as to whether the modification time matches the creation time. Upon an item being created either as a new or copy of another item, the modification time and creation time may be exactly the same, or slightly different depending on the configuration of the operating system or other application affecting the working version 50. If the determination from step 960 is positive, step 970 provides that the item is created as a result of an operation to create a new item.

If the determination from step 960 is negative, step 980 provides that the item is edited, new and possibly also a copy. Thus, step 980 provides for two possibilities. At this point, the modification time and creation time cannot be used to distinguish between the two possibilities. To resolve between the two possibilities, an embodiment may provide that all items identified in step 980 also be subjected to one or more steps of content matching. An algorithm may be used to compare contents of all files in step 980 with contents of other files identified as being new in working version 50, for purpose of determining if a file is new and edited, or new, copied and edited. The premise may be that the latter would have contents similar to another item identified as being new.

E. User-Interface

FIG. 10 illustrates a user-interface 1000 for use with an embodiment of the invention. The user-interface 1000 enables users to elect between same items changed on different computers. For example, with reference to FIG. 1, a user on first terminal 10 may perform operations on working version items downloaded from file system 40. The file system 40 may be shared, so other users may access it over the network. The other users may operate on an item in file system 40, while a user on first terminal 10 may operate on the corresponding working version item. When the synchronization request is made, a conflict may be presented. The file system item corresponding to the altered item

WO 02/097623

PCT/US02/16481

of the working version has been changed by another user who has accessed file system 40.

An embodiment of the invention enables the user making the synchronization request to elect between items in file system 40 and items in corresponding working version 50. An embodiment also allows the user making the request to choose how to decide conflicts between file system items updated by other computers and by working version items on the computer making the synchronization request.

The user-interface 1000 includes a first column 1110 and a second column 1120. The first column provides information about delta items on first computer 10. The second column 1120 provides information about delta item of file system 40. The delta items of the file system 40 may be identified by comparing the updated file systems with comparison file 70. A first portion 1125 of first column 1110 identifies the delta items of working version 50. A first portion of second column 1120 identifies the delta items of file system 40, as updated by other users. A second segment 1118 of first column 1110 identifies the operation or equivalent compound operations performed on the delta item of working version 50. Likewise, a second segment 1128 of second column 1120 identifies the operation or equivalent compound operation performed on the delta item of the updated file system. The operation(s) listed in second segment 1128 are assumed to have been performed by others who access the shared file system 40.

For each delta item listed in first column 1110 and second column 1120, the user may elect to keep the changes or maintain the item as it is on file system 40. If the delta item listed in first column 1110 conflicts with a delta item in second column 1120, the user may decide how to resolve the conflict. For example, an item in file system 40 may be downloaded onto working version 50, and subsequently operated on in working version 50. The same item downloaded may be accessed by another computer and operated in a different manner. When the synchronization request is made, the computer making that requested is presented with a conflict. The user of that computer may be given the ability to resolve the conflict. The user can pick which delta item to keep and use that item when reconciling with file system 40.

F. Hardware Description

Figure 11 is a block diagram that illustrates a computer system 1100 upon which an embodiment of the invention may be implemented. Computer system 1100 includes a bus 1102 or other communication mechanism for communicating information, and a processor 1104 coupled with bus 1102 for processing information. Computer system 1100 also includes a main memory 1106, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 1102 for storing information and instructions to be executed by processor 1104. Main memory 1106 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 1104. Computer system 1100 further includes a read only memory (ROM) 1108 or other static storage device coupled to bus 1102 for storing static information and instructions for processor 1104. A storage device 1110, such as a magnetic disk or optical disk, is provided and coupled to bus 1102 for storing information and instructions.

Computer system 1100 may be coupled via bus 1102 to a display 1112, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 1114, including alphanumeric and other keys, is coupled to bus 1102 for communicating information and command selections to processor 1104. Another type of user input device is cursor control 1116, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 1104 and for controlling cursor movement on display 1112. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

The invention is related to the use of computer system 1100 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 1100 in response to processor 1104 executing one or more sequences of one or more instructions contained in main memory 1106. Such instructions may be read into main memory 1106 from another computer-readable medium, such as storage device 1110. Execution of the sequences of instructions contained in main memory 1106 causes processor 1104 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

The term "computer-readable medium" as used herein refers to any medium that participates in providing instructions to processor 1104 for execution. Such a medium may

WO 02/097623

PCT/US02/16481

take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 1110. Volatile media includes dynamic memory, such as main memory 1106. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 1102. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 1104 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 1100 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 1102. Bus 1102 carries the data to main memory 1106, from which processor 1104 retrieves and executes the instructions. The instructions received by main memory 1106 may optionally be stored on storage device 1110 either before or after execution by processor 1104.

Computer system 1100 also includes a communication interface 1118 coupled to bus 1102. Communication interface 1118 provides a two-way data communication coupling to a network link 1120 that is connected to a local network 1122. For example, communication interface 1118 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 1118 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 1118 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

WO 02/097623

PCT/US02/16481

Network link 1120 typically provides data communication through one or more networks to other data devices. For example, network link 1120 may provide a connection through local network 1122 to a host computer 1124 or to data equipment operated by an Internet Service Provider (ISP) 1126. ISP 1126 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 1128. Local network 1122 and Internet 1128 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 1120 and through communication interface 1118, which carry the digital data to and from computer system 1100, are exemplary forms of carrier waves transporting the information.

Computer system 1100 can send messages and receive data, including program code, through the network(s), network link 1120 and communication interface 1118. In the Internet example, a server 1130 might transmit a requested code for an application program through Internet 1128, ISP 1126, local network 1122 and communication interface 1118.

The received code may be executed by processor 1104 as it is received, and/or stored in storage device 1110, or other non-volatile storage for later execution. In this manner, computer system 1100 may obtain application code in the form of a carrier wave.

G. Alternative Embodiments

While embodiments provided herein (see e.g. FIG. 1) describe reconciliation information as being in the form of metadata information, other embodiments may use portions or all of the content for items in first portion 46 (FIG. 1) to identify changes to items of the working version 50 (FIG. 1). In particular, content matching may be used to determine whether one item was copied from another. An intelligent algorithm may be employed to detect similarities between contents of items using the assumption that items with specific types of similarities are copies of one another.

Content matching may also be used as an additive step in a process such as described with FIG. 9. For example, if an equivalent operation is detected as shown by step 970, it may not be possible to determine whether the item was new, edited and also copied. Content matching may be required to detect whether one item is an edited copy of another item that was new.

Another use for content matching is as a tie-breaker, in case one or both time values of an item are exactly the same as another item. Considering the significant digits (i.e. one millionth of a second) of time values applied in popular operating systems such

WO 02/097623

PCT/US02/16481

as WIDNOWS, the chances of two items having exactly the same creation time or modification times is remote. However, if there is an exact match between time values of different items, embodiments of the invention allow for content matching to decipher between the two items.

While embodiments of the invention have been described for synchronizing files operated on different computers, it should be noted that other embodiments may be applied to stand alone or singular computer systems. For example, one application for an embodiment of the invention is to synchronize one file containing multiple entries with a backup file that was created as an archive. No interaction with other computer systems may be needed.

In some applications, it may be more useful to not detect certain equivalent compound operations, but rather assume more simple operations were performed on an item. Alternatively, the equivalent compound operations may be detected, but other operations may be used to update the file system 40. For example, an embodiment of the invention may treat the equivalent compound operation of $(N_x)C_y$ as N_x and N_y . Thus, during synchronization, file system 40 will be instructed to add two new items. Similarly, the compound operation of $E(xC_y)$ may be treated as N_y , where file system 40 may be instructed to create one new file, rather than copy X to Y , then edit it.

H. Conclusion

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

WO 02/097623

PCT/US02/16481

CLAIMS

What is claimed is:

1. A method for managing files, the method comprising:
mapping information about a file system to a comparison file, the
information identifying at least a first item in the file system located
at a first location, the first location being identifiable by a first
location information;
making a working version of a portion of the file system, the working
version including at least a first working item that corresponds to the
first item, the first working item initially being located at a second
location identifiable by the first location information; and
using information in the comparison file and about the working version to
subsequently determine if the first working item is located at a third
location identifiable by a second location information, the second
location information being different than the first location
information.
2. The method of claim 1, wherein if the first working item is located at
the third location, the method includes causing the first item in the
file system to move to a fourth location identifiable by the second
location information.
3. The method of claim 1, using information in the comparison file and about the
working version to subsequently determine if the first working item is located at a
third location identifiable by a second location information includes determining if
the first working item has a new name.
4. The method of claim 3, wherein making a working version of a portion of the file
system includes recording a creation time of the first working item in the
comparison file.
5. The method of claim 3, wherein using information in the comparison file and
about the working version to subsequently determine if the first working item is

WO 02/097623

PCT/US02/16481

located at a third location identifiable by a second location information includes locating the first working item in the working version using the creation time.

6. The method of claim 1, further including signaling the file system to delete the first item if the first working item is omitted from the working version.
7. The method of claim 6, further including using the creation time of the first working item to determine that the first working item has been deleted from the working version.
8. The method of claim 4, wherein making a working version of a portion of the file system includes recording a modification time of the first working item in the comparison file.
9. The method of claim 1, further comprising determining if the first working item was edited subsequent to making the working version.
10. The method of claim 8, further comprising determining if the first working item was edited subsequent to making the working version using the modification time of the first working item.
11. The method of claim 10, further comprising determining if a subsequent modification time of the first working item is different than the recorded modification time for the first working item.
12. A method for managing files, the method comprising:
mapping information about a file system to a comparison file;
making a working version of a portion of the file system; and
using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second working item in the working version.
13. The method of claim 12, further comprising causing the portion of the file system to include the first working item after determining the first working item was once copied from the second working item.

WO 02/097623

PCT/US02/16481

14. The method of claim 12, wherein the second working item originates from a first item in the portion of the file system.
15. The method of claim 13, wherein the second working item is created as new after the working version is made.
16. The method of claim 13, wherein a content of the first working item is different than a content of the second working item.
17. The method of claim 14, wherein making a working version includes recording both a creation time and a modification time in the comparison file for the second working item.
18. The method of claim 17, wherein using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second working item in the working version includes identifying a creation time and a modification time for the first working item.
19. The method of claim 18, wherein using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second working item in the working version includes detecting that the modification time of the first working item is before the creation time of the first working item.
20. The method of claim 19, wherein using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second working item in the working version includes matching the modification time of the first working item with the modification time of the second working item.
21. The method of claim 12, further comprising determining if the first working item was edited after being copied from the second working item.

WO 02/097623

PCT/US02/16481

22. The method of claim 12, further comprising determining that the first working item is new if the creation time of the first working item is different than a creation time of all of the items identified by the comparison file, and if the modification time for the first working item is greater than or equal to creation time for the first working item.
23. The method of claim 22, further comprising causing the portion of the file system to include the first working item after determining the first working item is new.
24. The method of claim 12, wherein using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second working item in the working version includes comparing a content of the first working item to a content of the second working item.
25. A method for managing files, the method comprising:
mapping information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identifiable by a first location information;
making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a second location, the second location being identifiable by the first location information; and
using information in the comparison file and about the working version to detect if at least one of two operations were performed on the working version, the operations including changing the first location information for the first working item, and editing a content of the first working item.
26. The method of claim 25, wherein making a working version of a portion of the file system includes recording an initial modification time for the first working item in the comparison file, the initial modification time recording a last instance when the first working item was either edited or created.

WO 02/097623

PCT/US02/16481

27. The method of claim 26, wherein detecting the operation of editing the content of the first working item includes subsequently determining if the initial modification time was changed.
28. The method of claim 27, further comprising causing the first item of the file system to include the edited content of the first working item.
29. The method of claim 25, wherein making a working version of the file system includes recording a creation time for the first working item.
30. The method of claim 29, wherein using information in the comparison file and about the working version to detect if at least one of two operations were performed on the working version includes locating the first working item in the working version using the creation time.
31. A method for managing files, the method comprising:
mapping information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identifiable by a first location information;
making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a second location, the second location being identifiable by the first location information; and
using information in the comparison file and about the working version to determine if a compound operation was performed on the first working item, the compound operation including at least two successive operations from a set of operations that consist of changing the first location information for the first working item, making a first working copy from the first working item, and editing a content of the first working item.
32. The method of claim 31, further comprising causing the portion of the file system to incorporate changes made by operations performed on the working version so that the first item matches the first working item.

WO 02/097623

PCT/US02/16481

33. The method of claim 31, wherein for one of the at least two compound operations being making a first working copy from the first working item, the method further includes determining if a selected working item in the working version was once copied from the first working item.
34. A computer system comprising:
a memory to store a working version of a portion of a file system located on a second computer system; and
a processor that marks at least a creation time of a first working item in the working version, the first working item originating from a first item of the portion of the file system, the processor subsequently using the creation time to detect an operation performed on the first working item.
35. The computer system of claim 34, further comprising a network interface to exchange communications over a network with the second computer, the communications being for generating the working version.
36. The computer system of claim 34, wherein the processor uses the creation time to determine if a location of the first working item was subsequently changed.
37. The computer system of claim 34, wherein the processor uses the creation time to determine if multiple operations were performed on the first working item.
38. The computer system of claim 34, wherein the processor initially marks a modification time with the creation time, the modification time being altered when the first item is subsequently edited.
39. The computer system of claim 38, wherein the processor compares the creation time to the modification time to identify the operation performed on the first working item.
40. The computer system of claim 38, wherein the processor detects that a content of the first working item was edited if the modification time is subsequently changed from initially being marked.

WO 02/097623

PCT/US02/16481

41. The computer system of claim 38, wherein the processor detects whether a second working item has the modification time of the first working item in order to determine that the second working item was once copied from the first working item.
42. The computer system of claim 38, wherein the processor detects the creation time of the first working item to determine that the first working item was moved to a new location, and wherein the processor detects that the modification time of the first working item has changed to determine that the first working item was also edited.
43. A computer system comprising:
a memory to store a working version of a portion of a file system accessible on a second computer, the working version including a plurality of working items, the memory also storing a comparison file including information about the plurality of working items; and
a processor configured to record at least a first time value for a first set of working items in the working version when the first set of working items are initially stored in the memory, the processor subsequently identifying a first time value of a second working item in the working version, the processor comparing the first time value of the second working item with the first time values of the first set of working items to detect an operation performed on the working version that affects the second working item.
44. The computer system of claim 43, further comprising a network interface to exchange communications over a network with the second computer, the communications being for generating the working version from the file system.
45. The computer system of claim 43, wherein the processor records the first time values of the first set of working items in a comparison file.
46. The computer system of claim 44, wherein the comparison file also includes information about a plurality of items in the portion of the file system, and wherein at least some of the plurality of working items originate from the plurality of items in the portion of the file system.

WO 02/097623

PCT/US02/16481

47. The computer system of claim 43, wherein the first time values of the first working items correspond to creation times for those items.
48. The computer system of claim 43, wherein the operation detected includes creating the second item after the first time values of the first set of working items are recorded.
49. The computer system of claim 48, wherein the operation detected includes creating the second item as a copy of one of the first set of working items.
50. The computer system of claim 43, wherein the processor records at least a second time value for each of the first set of working items when the first set of working items are initially stored in the memory, the second time values corresponding to modification times in which a content of the corresponding item was last altered or created.
51. The computer system of claim 50, wherein the processor identifies a second time value of a second working item, and compares the first time value and the second time value of the second working item to the first time value and the second time value of the first set of working items to identify the operation.
52. The computer system of claim 51, wherein the first set of working items originate from corresponding items in the portion of the file system, and wherein the processor detects the operation as creating the second working item after the working version is received.
53. The computer system of claim 52, wherein the processor detects the operation as creating the second working item from one of the first set of working items.
54. The computer system of claim 53, wherein the processor detects the operation as creating the second working item as a new item.
55. The computer system of claim 53, wherein the first time values of the first working items correspond to creation times for those items, and wherein the

WO 02/097623

PCT/US02/16481

second time values for the second working item and for the first set of working items correspond to a modification time for the respective working items in the working version.

56. The computer system of claim 55, wherein the operation is detected as creating the second working item from one of the first set of working items if the second time value matches the first time value of one of the first set of working items.
57. The computer system of claim 55, wherein multiple operations are detected, including a first operation creating a third working item, and a second operation creating the second working item as a copy of the third working item.
58. The computer system of claim 57, wherein the processor detects the first operation as creating the third working item as a new item if a first time value of the third working item does not match the first time value of one of the first set of working items.
59. The computer system of claim 58, wherein the processor detects the second operation as creating the second working item as a copy of the third working item if the second time value of the second working item is before the first time value of the second working item, and if the second time value of the second working item matches the first time value of the third working item.
60. A computer system comprising:
 - a server that stores a file system; and
 - a client coupleable to the file system to retrieve at least a portion of the file system, the portion of the file system being made available on the client as a working version, the working version initially including a plurality of working items, each working item originating from a corresponding item in the file system; and
 - processing resources to record initial information about the working version, and to subsequently detect two or more operations performed on the working version, the operations being selected from a group of operations consisting of adding an item to the working version, editing an item in the

WO 02/097623

PCT/US02/16481

working version, deleting an item from the working version, and moving an item in the working version to a new location.

61. The computer system of claim 60, wherein the processing resources signal the file system to change at least some of the file system to reflect the operations performed on the working version.

62. The computer system of claim 60, wherein the operation of adding an item to the working version includes copying a second working item from a first working item in the working version.

63. The computer system of claim 60, wherein the operation of adding an item to the working version includes creating a second working item as a new item.

64. The computer system of claim 60, wherein the initial information recorded about the working file includes a first time value for a plurality of working items in the working version, the first time value indicating an origination of each working item from either another working item in the plurality of working items or from another item on the server.

65. The computer system of claim 64, wherein the first time value is a creation time corresponding to when that item was created on the client.

66. The computer system of claim 64, wherein the first time value is a modification time that records a last instance of when that item was altered or created.

67. The computer system of claim 60, wherein the processing resources are located on the client.

68. The computer system of claim 60, wherein the processing resources are located on the server.

WO 02/097623

PCT/US02/16481

69. The computer system of claim 59, wherein the processing resources are shared between the client and the server.

70. The computer system of claim 60, wherein the file system is shared with a plurality of clients, each of the plurality of clients being able to alter the file system, and wherein an altered version of the file system is signaled to the client upon the client making a request to compare the working version with the file system.

71. The computer system of claim 70, wherein the server signals the altered version of the file system to the client, and wherein the processing resources are on the client to detect one or more operations performed on the file system after the processing resources record the initial information about the working version.

72. The computer system of claim 71, wherein the processing resources compare the altered version of the file system to the initial information recorded about the working version to detect the one or more operations performed on the file system.

73. The computer system of claim 72, wherein the processing resources signal the user to select whether to update the file system with each working version item that was altered or created as a result of one or more of the operations.

74. The computer system of claim 73, wherein the processing resources signal the user to select whether to update the file system with each item in the altered file system that was created or altered as a result of one or more operations performed on the file system.

75. The computer system of claim 74, wherein the processing resources signal the user to select between a first item of the working version and a first item of the altered file system, the first item of the working version being subject to a first operation from the group of operations, the first item of the file system being subject to a second operation from the group of

WO 02/097623

PCT/US02/16481

operations, wherein the first item of the working version and the first item of the altered file system each originate from a first item in the file system, and wherein the first operation is different than the second operation.

76. A computer readable medium carrying instructions for managing files, the instructions including instructions for performing the steps of:

mapping information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identified by a first location information;

making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a second location by the first location information; and

using information in the comparison file and about the working version to determine if the first working item is located at a third location identified by a second location information, the second location information being different than the first location information.

77. The computer readable medium of claim 76, wherein if the first working item is located at the third location, the instructions of computer readable medium includes instructions for causing the first item in the file system to move to a fourth location identified by the second location information.

78. The computer readable medium of claim 76, wherein the step of using information in the comparison file and about the working version to determine if the first working item is subsequently located at a third location identifiable by a second location information includes instructions for determining if the first working item has a new name.

79. The computer readable medium of claim 78, wherein the step of making a working version of a portion of the file system includes instructions for recording a creation time of the first working item in the comparison file.

WO 02/097623

PCT/US02/16481

80. The computer readable medium of claim 78, wherein the step of using information in the comparison file and about the working version to determine if the first working item is subsequently located at a third location identifiable by a second location information includes instructions for locating the first working item in the working version using the creation time.
81. The computer readable medium of claim 76, further including instructions for signaling the file system to delete the first item if the first working item is omitted from the working version.
82. The computer readable medium of claim 81, further including instructions for using the creation time of the first working item to determine that the first working item has been deleted from the working version.
83. The computer readable medium of claim 79, wherein the step of making a working version of the file system includes instructions for recording a modification time of the first working item in the comparison file.
84. The computer readable medium of claim 76, further comprising instructions for determining if the first working item was edited subsequent to making the working version.
85. The computer readable medium of claim 83, further comprising instructions for determining if the first working item was edited subsequent to making the working version using the modification time of the first working item.
86. The computer readable medium of claim 85, wherein the step of using the modification time of the first working item includes instructions for determining if a subsequent modification time of the first working item is different than the recorded modification time for the first working item.
87. A computer readable medium carrying instructions for managing files on different computers, the instructions including instructions for performing the steps of: mapping information about a file system to a comparison file

WO 02/097623

PCT/US02/16481

making a working version of a portion of the file system;
using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second working item in the working version.

88. The computer readable medium of claim 87, further comprising instructions for causing the file system to include the first working item after determining the first working item was once copied from the second working item.
89. The computer readable medium of claim 87, wherein the second working item originates from a first item in the file system.
90. The computer readable medium of claim 88, wherein a content of the first working item is different than a content of the second working item.
91. The computer readable medium of claim 89, wherein the step of making a working version includes instructions for recording both a creation time and a modification time in the comparison file for the second working item.
92. The computer readable medium of claim 91, wherein the step of using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second working item in the working version includes instructions for identifying a creation time and a modification time for the first working item.
93. The computer readable medium of claim 92, wherein the step of using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second working item in the working version includes instructions for detecting that the modification time of the first working item is before the creation time of the first working item.
94. The computer readable medium of claim 93, wherein the step of using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second

WO 02/097623

PCT/US02/16481

working item in the working version includes instructions for matching the modification time of the first working item with the modification time of the second working item.

95. The computer readable medium of claim 89, further comprising instructions for performing the steps of determining if the first working item was edited after being copied from the second working item.
96. The computer readable medium of claim 87, further comprising instructions for performing the steps of determining that the first working item is a new file if the creation time of the first working item is different than a creation time of all of the items identified by the comparison file, and if the modification time for the first working item is greater than or equal to the creation time for the first working item.
97. The computer readable medium of claim 96, further comprising instructions for performing the step of causing the file system to include the first working item after determining the first working item is new.
98. The computer readable medium of claim 82, wherein the step of using information in the comparison file and about the working version to subsequently determine if a first working item in the working version was once copied from a second working item in the working version includes instructions for comparing a content of the first working item to a content of the second working item.
99. A computer readable medium for managing files, the computer readable medium carrying instructions including instructions for performing the steps of: mapping information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identified by a first location information; making a working version of a portion of the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a second location, the second location being identified by the first location information; and

WO 02/097623

PCT/US02/16481

- using information in the comparison file and about the working version to detect if at least one of two operations were performed on the working version, the operations including changing the first location information for the first working item, and editing a content of the first working item.
100. The computer readable medium of claim 99, wherein the step of making a working version of a portion of the file system includes instructions for recording an initial modification time for the first working item in the comparison file, the initial modification time recording a last instance when the first working item was edited or created.
101. The computer readable medium of claim 100, wherein the step of using information in the comparison file and about the working version to detect if at least one of two operations were performed on the working version includes instructions for subsequently determining if the initial modification time was changed.
102. The computer readable medium of claim 101, further comprising instructions for performing the step of causing the file system to include the edited content of the first working item.
103. The computer readable medium of claim 99, wherein the step of making a working version of a portion of the file system includes instructions for recording a creation time for the first working item.
104. The computer readable medium of claim 103, wherein the step of using information in the comparison file and about the working version to detect if at least one of two operations were performed on the working version includes instructions for locating the first working item in the working version using the creation time.
105. A computer readable medium carrying instructions for managing files, the instructions including instructions for performing the steps of:

WO 02/097623

PCT/US02/16481

mapping information about a file system to a comparison file, the information identifying at least a first item in the file system located at a first location, the first location being identified by a first location information;

making a working version of a portion the file system, the working version including at least a first working item originating from the first item, the first working item initially being located at a second location, the second location being identified by the first location information; and

using information in the comparison file and about the working version to determine if a compound operation was performed on the first working item, the compound operation including at least two successive operations from a set of operations that consist of changing the first location information for the first working item, making a first working copy from the first working item, and editing a content of the first working item.

106. The computer readable medium of claim 105, further comprising instructions for indicating which of the operations were performed on the first working item to change the first item in the file system if the compound operation is determined to have been performed.

107. The computer readable medium of claim 105, wherein for one of the at least two compound operations being making a first working copy from the first working item, the computer readable medium further includes instructions for determining if a selected working item in the working version was once copied from the first working item.

WO 02/097623

1/11

PC7/US02/16481

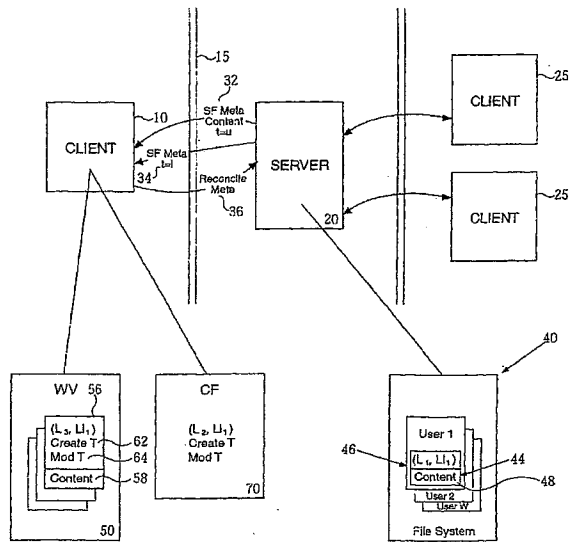


FIGURE 1

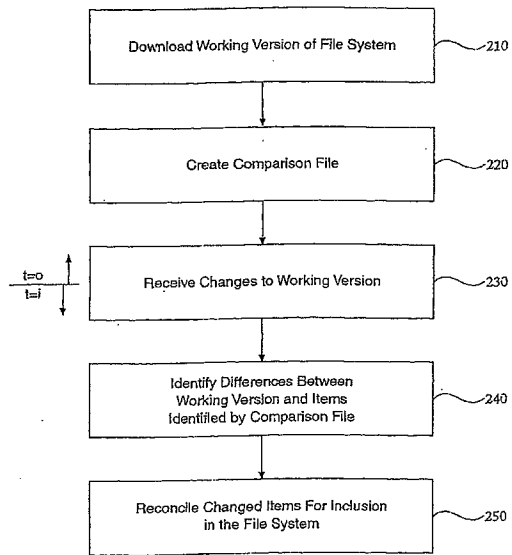


FIGURE 2

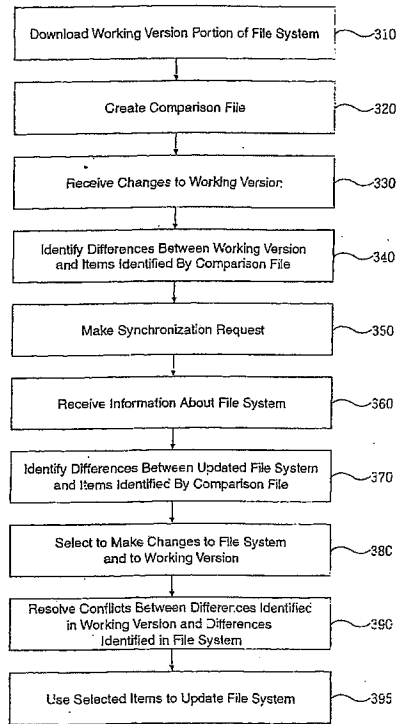


FIGURE 3

WO 02/097623

4/11

PCT/US02/16481

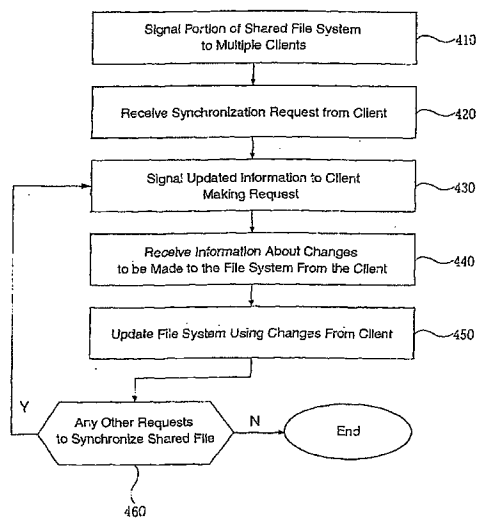


FIGURE 4

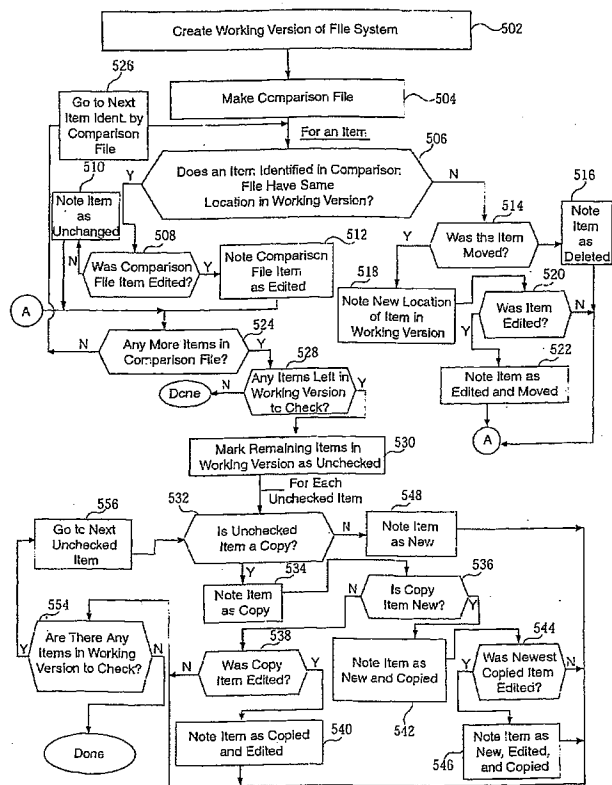


FIGURE 5

WO 02/097623

6/11

PCT/US02/16481

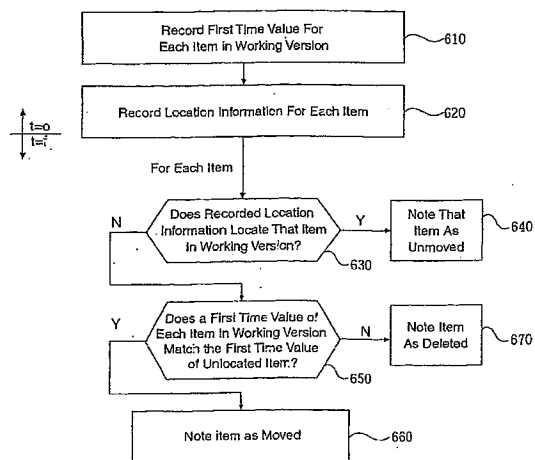


FIGURE 6

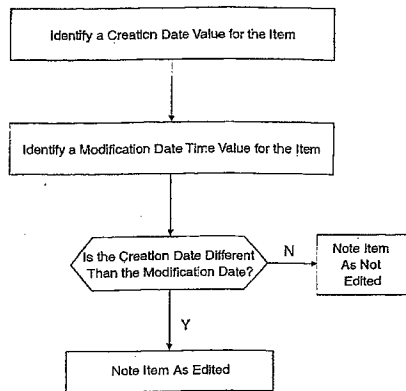


FIGURE 7

WO 02/097623

8/11

PCT/US02/16481

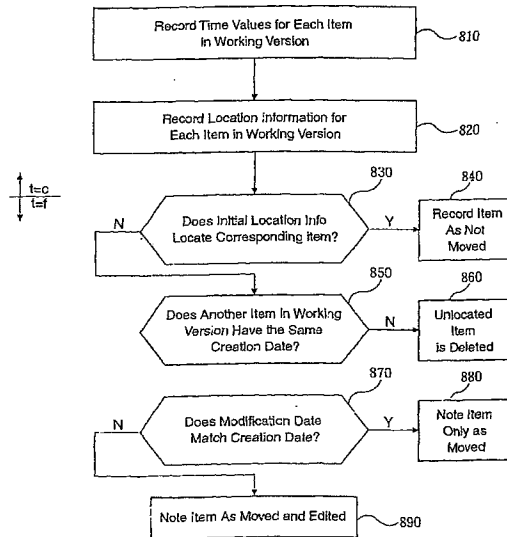


FIGURE 8

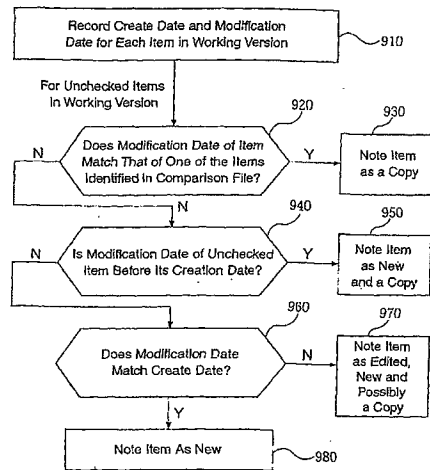


FIGURE 9

Computer A		Computer B	
File Name	Action Required	File Name	Action Required
A ₁	₁ M _y	B ₁	_A C _B
A ₂	E	B ₂	E
•	•	•	•
•	•	•	•
•	•	•	•
A _n	(N) _x C _y	B _n	D

USER INTERFACE

FIGURE 10

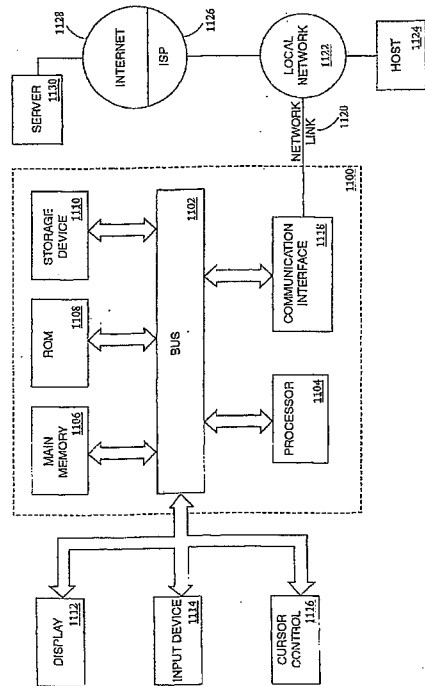


FIGURE 11

【国際公開パンフレット（コレクトバージョン）】

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
5 December 2002 (05.12.2002)

PCT

(10) International Publication Number
WO 02/097623 A3

(51) International Patent Classification: G06F 9/46, 9/44

(21) International Application Number: PCT/US02/16481

(22) International Filing Date: 24 May 2002 (24.05.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data: 0208661-3 25 May 2001 (25.05.2001) US

(71) Applicant: ORACLE INTERNATIONAL CORPORATION (US); 500 Oracle Parkway, Redwood Shores, CA 94065 (US)

(72) Inventor: KIRNOS, Ilya; 1729 Woodland Avenue, Unit M, Palo Alto, CA 94303 (US)

(74) Agents: MAHAMED, Van et al.; HICKMAN PALERMO TRUONG & BECKER, LLP, 1600 Willow Street, San Jose, CA 95125 (US)

(81) Designated States (national): AU, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GR, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SI, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NI, SN, TD, TG).

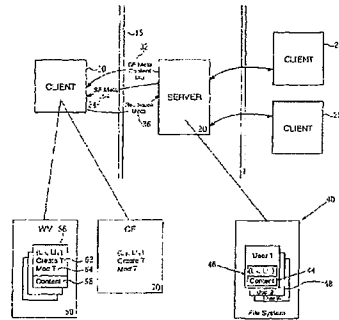
Published:

with international search report

(88) Date of publication of the international search report: 2 April 2003

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: MANAGEMENT AND SYNCHRONIZATION APPLICATION FOR NETWORK FILE SYSTEM



WO 02/097623 A3

(57) Abstract: An application is provided for managing files. Information about a file system is mapped to a companion file. A working version is made from the file system. The application identify one or more operations were performed on the working version. The identified operations are recreated on the file system during a synchronization operation.

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International Application No. PCT/US 02/16481
A. CLASSIFICATION OF SUBJECT MATTER IPC 7 606F9/46 606F9/44		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC 7 606F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal, INSPEC, IBM-TDB		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 903 725 A (COLYER ADRIAN MARK) 11 May 1999 (1999-05-11) column 6, line 46 -column 8, line 11	1,12,25, 31,34, 43,60, 76,87, 99,105
A	EP 0 707 263 A (IBM) 17 April 1996 (1996-04-17) column 2, line 29 -column 3, line 5 -/-	1,12,25, 31,34, 43,60, 76,87, 99,105
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in annex.		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document relevant to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to underpin the principle or theory underlying the invention "X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "Z" document member of the same patent family		
Date of the actual completion of the international search 17 December 2002		Date of mailing of the international search report 07/01/2003
Name and mailing address of the ISA European Patent Office, P.O. Box 5818, Palaisin 2 NL-2280 HV Rijswijk Tel. (+31-70) 910-2000, Telex 31 651 epo nl, Fax: (+31-70) 940-0918		Authorized officer Brandt, J

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT		International Application No. PCT/US 02/16481
C (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>MENON S ET AL: "OBJECT REPLACEMENT USING DYNAMIC PROXY UPDATES" PROCEEDINGS OF THE INTERNATIONAL WORKSHOP ON CONFIGURABLE DISTRIBUTED SYSTEMS, IEEE COMPUTER SOCIETY, LOS ALAMITOS, CA, US, 1994, pages 82-91, XP002004310 page 82, right-hand column, line 19 - line 29 page 84, left-hand column, line 46 - line 52 page 84, right-hand column, line 1 - page 86, left-hand column, line 9 page 86, right-hand column, line 31 - line 54 page 87, left-hand column, line 8 - line 34 page 88, left-hand column, line 1 - page 89, left-hand column, line 2</p>	<p>1,12,25, 31,34, 43,60, 76,87, 99,105</p>

INTERNATIONAL SEARCH REPORT Information on patent family members				International Application No. PCT/US 02/16481	
Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 5903725	A	11-05-1999	GB 2305271 A		02-04-1997
			DE 69607851 D1		25-05-2000
			EP 0850445 A1		01-07-1998
			WO 9710545 A1		20-03-1997
			JP 10511202 T		27-10-1998
EP 0707263	A	17-04-1996	US 5745766 A		28-04-1998
			EP 0707263 A1		17-04-1996
			JP 8095756 A		12-04-1996

フロントページの続き

(81)指定国 AP(GH,GM,KE,LS,MW,MZ,SD,SL,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,MD,RU,TJ,TM),EP(AT, BE,CH,CY,DE,DK,ES,FI,FR,GB,GR,IE,IT,LU,MC,NL,PT,SE,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,ML,MR,NE,SN, TD,TG),AE,AG,AL,AM,AT,AU,AZ,BA,BB,BG,BR,BY,BZ,CA,CH,CN,CO,CR,CU,CZ,DE,DK,DM,DZ,EC,EE,ES,FI,GB,GD,GE, GH,GM,HR,HU,ID,IL,IN,IS,JP,KE,KG,KP,KR,KZ,LC,LK,LR,LS,LT,LU,LV,MA,MD,MG,MK,MN,MW,MX,MZ,NO,NZ,OM,PH,P L,PT,RO,RU,SD,SE,SG,SI,SK,SL,TJ,TM,TN,TR,TT,TZ,UA,UG,UZ,VN,YU,ZA,ZM,ZW

(74)代理人 100098316

弁理士 野田 久登

(74)代理人 100109162

弁理士 酒井 将行

(72)発明者 キルノス, イルヤ

アメリカ合衆国、9 4 3 0 3 カリフォルニア州、パロ・アルト、ウッドランド・アベニュー、1 7
2 9、ユニット・エム

Fターム(参考) 5B082 GA05 GB01 GB06 HA03

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第6部門第3区分

【発行日】平成18年1月5日(2006.1.5)

【公表番号】特表2005-507100(P2005-507100A)

【公表日】平成17年3月10日(2005.3.10)

【年通号数】公開・登録公報2005-010

【出願番号】特願2003-500739(P2003-500739)

【国際特許分類】

G 0 6 F 12/00 (2006.01)

【F I】

G 0 6 F 12/00 5 3 3 F

G 0 6 F 12/00 5 1 7

【手続補正書】

【提出日】平成17年5月12日(2005.5.12)

【手続補正1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

ファイルを管理するための方法であって、

ファイルシステムについての情報を比較ファイルへマッピングするステップを含み、情報は、第1の位置に位置するファイルシステム内の少なくとも第1の項目を識別し、第1の位置は第1の位置情報により識別可能であり、前記方法はさらに、

ファイルシステムの一部の作業バージョンを作成するステップを含み、作業バージョンは第1の項目に対応する少なくとも第1の作業項目を含み、第1の作業項目は最初、第1の位置情報により識別可能な第2の位置に位置しており、前記方法はさらに、

比較ファイル内のおよび作業バージョンについての情報を用いて、第1の作業項目が第2の位置情報により識別可能な第3の位置に位置しているかどうかをその後判断するステップを含み、第2の位置情報は第1の位置情報とは異なる、方法。

【請求項2】

第1の作業項目が第3の位置に位置する場合、ファイルシステム内の第1の項目を、第2の位置情報により識別可能な第4の位置へ移動させるステップを含む、請求項1に記載の方法。

【請求項3】

比較ファイル内のおよび作業バージョンについての情報を用いて、第1の作業項目が第2の位置情報により識別可能な第3の位置に位置しているかどうかをその後判断するステップは、第1の作業項目が新しい名前を有しているかどうかを判断するステップを含む、請求項1に記載の方法。

【請求項4】

ファイルシステムの一部の作業バージョンを作成するステップは、第1の作業項目の作成時刻を比較ファイルに記録するステップを含む、請求項3に記載の方法。

【請求項5】

比較ファイル内のおよび作業バージョンについての情報を用いて、第1の作業項目が第2の位置情報により識別可能な第3の位置に位置しているかどうかをその後判断するステップは、作業バージョン内の第1の作業項目の位置を、作成時刻を用いて特定するステップを含む、請求項3に記載の方法。

【請求項 6】

第 1 の作業項目が作業バージョンから省略されている場合、第 1 の項目を削除するようファイルシステムに信号を送るステップをさらに含む、請求項 1 に記載の方法。

【請求項 7】

第 1 の作業項目の作成時刻を用いて、第 1 の作業項目が作業バージョンから削除されたと判断するステップをさらに含む、請求項 6 に記載の方法。

【請求項 8】

ファイルシステムの一部の作業バージョンを作成するステップは、第 1 の作業項目の修正時刻を比較ファイルに記録するステップを含む、請求項 4 に記載の方法。

【請求項 9】

作業バージョンの作成後に第 1 の作業項目が編集されたかどうかを判断するステップをさらに含む、請求項 1 に記載の方法。

【請求項 10】

作業バージョンの作成後に第 1 の作業項目が編集されたかどうかを、第 1 の作業項目の修正時刻を用いて判断するステップをさらに含む、請求項 8 に記載の方法。

【請求項 11】

第 1 の作業項目のその後の修正時刻が、第 1 の作業項目についての記録された修正時刻と異なるかどうかを判断するステップをさらに含む、請求項 10 に記載の方法。

【請求項 12】

ファイルを管理するための方法であって、
ファイルシステムについての情報を比較ファイルへマッピングするステップと、
ファイルシステムの一部の作業バージョンを作成するステップと、
比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップとを含む、方法。

【請求項 13】

第 1 の作業項目がかつて第 2 の作業項目からコピーされたと判断した後で、ファイルシステムの一部に、第 1 の作業項目を含めさせるステップをさらに含む、請求項 12 に記載の方法。

【請求項 14】

第 2 の作業項目は、ファイルシステムの一部における第 1 の項目から生じている、請求項 12 に記載の方法。

【請求項 15】

作業バージョンが作成された後で、第 2 の作業項目が新規として作成される、請求項 13 に記載の方法。

【請求項 16】

第 1 の作業項目の内容は第 2 の作業項目の内容と異なっている、請求項 13 に記載の方法。

【請求項 17】

作業バージョンを作成するステップは、第 2 の作業項目について作成時刻および修正時刻の双方を比較ファイルに記録するステップを含む、請求項 14 に記載の方法。

【請求項 18】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップは、第 1 の作業項目についての作成時刻および修正時刻を識別するステップを含む、請求項 17 に記載の方法。

【請求項 19】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップは、第 1 の作業項目の修正時刻が第 1 の作業項目の作成時刻の前

であることを検出するステップを含む、請求項 18 に記載の方法。

【請求項 20】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップは、第 1 の作業項目の修正時刻を第 2 の作業項目の修正時刻と整合させるステップを含む、請求項 19 に記載の方法。

【請求項 21】

第 1 の作業項目が第 2 の作業項目からコピーされた後で編集されたかどうかを判断するステップをさらに含む、請求項 12 に記載の方法。

【請求項 22】

第 1 の作業項目の作成時刻が比較ファイルにより識別された全項目の作成時刻と異なる場合、および第 1 の作業項目についての修正時刻が第 1 の作業項目についての作成時刻よりも大きいまたはそれと等しい場合、第 1 の作業項目は新規であると判断するステップをさらに含む、請求項 12 に記載の方法。

【請求項 23】

第 1 の作業項目が新規であると判断した後で、ファイルシステムの一部に、第 1 の作業項目を含めさせるステップをさらに含む、請求項 22 に記載の方法。

【請求項 24】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第 1 の作業項目がかつて作業バージョン内の第 2 の作業項目からコピーされたかどうかをその後判断するステップは、第 1 の作業項目の内容を第 2 の作業項目の内容と比較するステップを含む、請求項 12 に記載の方法。

【請求項 25】

ファイルを管理するための方法であって、

ファイルシステムについての情報を比較ファイルへマッピングするステップを含み、情報は、第 1 の位置に位置するファイルシステム内の少なくとも第 1 の項目を識別し、第 1 の位置は第 1 の位置情報により識別可能であり、前記方法はさらに、

ファイルシステムの一部の作業バージョンを作成するステップを含み、作業バージョンは第 1 の項目から生じる少なくとも第 1 の作業項目を含み、第 1 の作業項目は最初、第 2 の位置に位置しており、第 2 の位置は第 1 の位置情報により識別可能であり、前記方法はさらに、

比較ファイル内のおよび作業バージョンについての情報を用いて、2つの動作のうちの少なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップを含み、動作は、第 1 の作業項目についての第 1 の位置情報の変更と、第 1 の作業項目の内容の編集とを含む、方法。

【請求項 26】

ファイルシステムの一部の作業バージョンを作成するステップは、第 1 の作業項目についての初期修正時刻を比較ファイルに記録するステップを含み、初期修正時刻は第 1 の作業項目が編集または作成された最後の時点を記録する、請求項 25 に記載の方法。

【請求項 27】

第 1 の作業項目の内容を編集する動作を検出するステップは、初期修正時刻が変更されたかどうかをその後判断するステップを含む、請求項 26 に記載の方法。

【請求項 28】

ファイルシステムの第 1 の項目に、第 1 の作業項目の編集された内容を含めさせるステップをさらに含む、請求項 27 に記載の方法。

【請求項 29】

ファイルシステムの作業バージョンを作成するステップは、第 1 の作業項目についての作成時刻を記録するステップを含む、請求項 25 に記載の方法。

【請求項 30】

比較ファイル内のおよび作業バージョンについての情報を用いて、2つの動作のうちの

少なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップは、作業バージョン内の第1の作業項目の位置を、作成時刻を用いて特定するステップを含む、請求項29に記載の方法。

【請求項31】

ファイルを管理するための方法であって、

ファイルシステムについての情報を比較ファイルへマッピングするステップを含み、情報は、第1の位置に位置するファイルシステム内の少なくとも第1の項目を識別し、第1の位置は第1の位置情報により識別可能であり、前記方法はさらに、

ファイルシステムの一部の作業バージョンを作成するステップを含み、作業バージョンは第1の項目から生じる少なくとも第1の作業項目を含み、第1の作業項目は最初、第2の位置に位置しており、第2の位置は第1の位置情報により識別可能であり、前記方法はさらに、

比較ファイル内のおよび作業バージョンについての情報を用いて、複合動作が第1の作業項目に対して実行されたかどうかを判断するステップを含み、複合動作は、第1の作業項目についての第1の位置情報の変更、第1の作業項目からの第1の作業コピーの作成、および第1の作業項目の内容の編集からなる一組の動作からの少なくとも2つの連続する動作を含む、方法。

【請求項32】

第1の項目が第1の作業項目に整合するように、ファイルシステムの一部に、作業バージョンに対して実行された動作によりなされた変更を取入れさせるステップをさらに含む、請求項31に記載の方法。

【請求項33】

少なくとも2つの複合動作のうちの1つが第1の作業項目から第1の作業コピーを作成するために、作業バージョン内の選択された作業項目がかつて第1の作業項目からコピーされたかどうかを判断するステップをさらに含む、請求項31に記載の方法。

【請求項34】

ファイルを管理するための命令を搬送するコンピュータ可読媒体であって、命令は、

ファイルシステムについての情報を比較ファイルへマッピングするステップを行なうための命令を含み、情報は、第1の位置に位置するファイルシステム内の少なくとも第1の項目を識別し、第1の位置は第1の位置情報により識別され、前記命令はさらに、

ファイルシステムの一部の作業バージョンを作成するステップを行なうための命令を含み、作業バージョンは第1の項目から生じる少なくとも第1の作業項目を含み、第1の作業項目は最初、第1の位置情報によって第2の位置に位置しており、前記命令はさらに、

比較ファイル内のおよび作業バージョンについての情報を用いて、第1の作業項目が第2の位置情報によって識別される第3の位置に位置しているかどうかを判断するステップを行なうための命令を含み、第2の位置情報は第1の位置情報とは異なる、コンピュータ可読媒体。

【請求項35】

第1の作業項目が第3の位置に位置する場合、コンピュータ可読媒体の命令は、ファイルシステム内の第1の項目を、第2の位置情報によって識別される第4の位置へ移動させるための命令を含む、請求項34に記載のコンピュータ可読媒体。

【請求項36】

比較ファイル内のおよび作業バージョンについての情報を用いて、第1の作業項目がその後、第2の位置情報により識別可能な第3の位置に位置しているかどうかを判断するステップは、第1の作業項目が新しい名前を有しているかどうかを判断するための命令を含む、請求項34に記載のコンピュータ可読媒体。

【請求項37】

ファイルシステムの一部の作業バージョンを作成するステップは、第1の作業項目の作成時刻を比較ファイルに記録するための命令を含む、請求項36に記載のコンピュータ可読媒体。

【請求項 38】

比較ファイル内のおよび作業バージョンについての情報を用いて、第1の作業項目がその後、第2の位置情報により識別可能な第3の位置に位置しているかどうかを判断するステップは、作業バージョン内の第1の作業項目の位置を、作成時刻を用いて特定するための命令を含む、請求項36に記載のコンピュータ可読媒体。

【請求項 39】

第1の作業項目が作業バージョンから省略されている場合、第1の項目を削除するようファイルシステムに信号を送るための命令をさらに含む、請求項34に記載のコンピュータ可読媒体。

【請求項 40】

第1の作業項目の作成時刻を用いて、第1の作業項目が作業バージョンから削除されたと判断するための命令をさらに含む、請求項39に記載のコンピュータ可読媒体。

【請求項 41】

ファイルシステムの作業バージョンを作成するステップは、第1の作業項目の修正時刻を比較ファイルに記録するための命令を含む、請求項37に記載のコンピュータ可読媒体。

【請求項 42】

作業バージョンの作成後に第1の作業項目が編集されたかどうかを判断するための命令をさらに含む、請求項34に記載のコンピュータ可読媒体。

【請求項 43】

作業バージョンの作成後に第1の作業項目が編集されたかどうかを、第1の作業項目の修正時刻を用いて判断するための命令をさらに含む、請求項41に記載のコンピュータ可読媒体。

【請求項 44】

第1の作業項目の修正時刻を用いるステップは、第1の作業項目のその後の修正時刻が第1の作業項目についての記録された修正時刻と異なるかどうかを判断するための命令を含む、請求項43に記載のコンピュータ可読媒体。

【請求項 45】

異なるコンピュータ上のファイルを管理するための命令を搬送するコンピュータ可読媒体であって、命令は、

ファイルシステムについての情報を比較ファイルへマッピングするステップと、

ファイルシステムの一部の作業バージョンを作成するステップと、

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップとを行なうための命令を含む、コンピュータ可読媒体。

【請求項 46】

第1の作業項目がかつて第2の作業項目からコピーされたと判断した後でファイルシステムに第1の作業項目を含めさせるための命令をさらに含む、請求項45に記載のコンピュータ可読媒体。

【請求項 47】

第2の作業項目は、ファイルシステムにおける第1の項目から生じている、請求項45に記載のコンピュータ可読媒体。

【請求項 48】

第1の作業項目の内容は第2の作業項目の内容と異なっている、請求項46に記載のコンピュータ可読媒体。

【請求項 49】

作業バージョンを作成するステップは、第2の作業項目について作成時刻および修正時刻の双方を比較ファイルに記録するための命令を含む、請求項47に記載のコンピュータ可読媒体。

【請求項 50】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップは、第1の作業項目についての作成時刻および修正時刻を識別するための命令を含む、請求項49に記載のコンピュータ可読媒体。

【請求項51】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップは、第1の作業項目の修正時刻が第1の作業項目の作成時刻の前であることを検出するための命令を含む、請求項50に記載のコンピュータ可読媒体。

【請求項52】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップは、第1の作業項目の修正時刻を第2の作業項目の修正時刻と整合させるための命令を含む、請求項51に記載のコンピュータ可読媒体。

【請求項53】

第1の作業項目が第2の作業項目からコピーされた後で編集されたかどうかを判断するステップを行なうための命令をさらに含む、請求項47に記載のコンピュータ可読媒体。

【請求項54】

第1の作業項目の作成時刻が比較ファイルにより識別された全項目の作成時刻と異なる場合、および第1の作業項目についての修正時刻が第1の作業項目についての作成時刻よりも大きいまたはそれと等しい場合、第1の作業項目は新規ファイルであると判断するステップを行なうための命令をさらに含む、請求項45に記載のコンピュータ可読媒体。

【請求項55】

第1の作業項目が新規であると判断した後でファイルシステムに第1の作業項目を含めさせるステップを行なうための命令をさらに含む、請求項54に記載のコンピュータ可読媒体。

【請求項56】

比較ファイル内のおよび作業バージョンについての情報を用いて、作業バージョン内の第1の作業項目がかつて作業バージョン内の第2の作業項目からコピーされたかどうかをその後判断するステップは、第1の作業項目の内容を第2の作業項目の内容と比較するための命令を含む、請求項40に記載のコンピュータ可読媒体。

【請求項57】

ファイルを管理するためのコンピュータ可読媒体であって、コンピュータ可読媒体は、ファイルシステムについての情報を比較ファイルへマッピングするステップを行なうための命令を含む命令を搬送しており、その情報は、第1の位置に位置するファイルシステム内の少なくとも第1の項目を識別し、第1の位置は第1の位置情報によって識別され、前記コンピュータ可読媒体はさらに、

ファイルシステムの一部の作業バージョンを作成するステップを行なうための命令を含む命令を搬送しており、作業バージョンは第1の項目から生じる少なくとも第1の作業項目を含み、第1の作業項目は最初、第2の位置に位置しており、第2の位置は第1の位置情報によって識別され、前記コンピュータ可読媒体はさらに、

比較ファイル内のおよび作業バージョンについての情報を用いて、2つの動作のうちの少なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップを行なうための命令を含む命令を搬送しており、動作は、第1の作業項目についての第1の位置情報の変更と、第1の作業項目の内容の編集とを含む、コンピュータ可読媒体。

【請求項58】

ファイルシステムの一部の作業バージョンを作成するステップは、第1の作業項目についての初期修正時刻を比較ファイルに記録するための命令を含み、初期修正時刻は第1の作業項目が編集または作成された最後の時点を記録する、請求項57に記載のコンピュータ可読媒体。

【請求項 5 9】

比較ファイル内のおよび作業バージョンについての情報を用いて、2つの動作のうちの少なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップは、初期修正時刻が変更されたかどうかをその後判断するための命令を含む、請求項 5 8 に記載のコンピュータ可読媒体。

【請求項 6 0】

ファイルシステムに第 1 の作業項目の編集された内容を含めさせるステップを行なうための命令をさらに含む、請求項 5 9 に記載のコンピュータ可読媒体。

【請求項 6 1】

ファイルシステムの一部の作業バージョンを作成するステップは、第 1 の作業項目についての作成時刻を記録するための命令を含む、請求項 5 7 に記載のコンピュータ可読媒体。

【請求項 6 2】

比較ファイル内のおよび作業バージョンについての情報を用いて、2つの動作のうちの少なくとも一つが作業バージョンに対して実行されたかどうかを判断するステップは、作業バージョン内の第 1 の作業項目の位置を、作成時刻を用いて特定するための命令を含む、請求項 6 1 に記載のコンピュータ可読媒体。

【請求項 6 3】

ファイルを管理するための命令を搬送するコンピュータ可読媒体であって、命令は、ファイルシステムについての情報を比較ファイルへマッピングするステップを行なうための命令を含み、情報は、第 1 の位置に位置するファイルシステム内の少なくとも第 1 の項目を識別し、第 1 の位置は第 1 の位置情報によって識別され、前記命令はさらに、

ファイルシステムの一部の作業バージョンを作成するステップを行なうための命令を含み、作業バージョンは第 1 の項目から生じる少なくとも第 1 の作業項目を含み、第 1 の作業項目は最初、第 2 の位置に位置しており、第 2 の位置は第 1 の位置情報によって識別され、前記命令はさらに、

比較ファイル内のおよび作業バージョンについての情報を用いて、複合動作が第 1 の作業項目に対して実行されたかどうかを判断するステップを行なうための命令を含み、複合動作は、第 1 の作業項目についての第 1 の位置情報の変更、第 1 の作業項目からの第 1 の作業コピーの作成、および第 1 の作業項目の内容の編集からなる一組の動作からの少なくとも2つの連続する動作を含む、コンピュータ可読媒体。

【請求項 6 4】

複合動作が実行されたと判断された場合、ファイルシステム内の第 1 の項目を変更するためにどちらの動作が第 1 の作業項目に対して実行されたかを示すための命令をさらに含む、請求項 6 3 に記載のコンピュータ可読媒体。

【請求項 6 5】

少なくとも2つの複合動作のうちの1つが第 1 の作業項目から第 1 の作業コピーを作成するために、作業バージョン内の選択された作業項目がかつて第 1 の作業項目からコピーされたかどうかを判断するための命令をさらに含む、請求項 6 3 に記載のコンピュータ可読媒体。